

О. С. Боровик
А. В. Парфенов
В. И. Сырямкин

УВЛЕКАТЕЛЬНЫЕ ИГРЫ НА БЫТОВОМ КОМПЬЮТЕРЕ

АН08

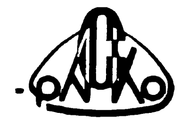
Малое предприятие «Раско»



Основана в 1947 году
Выпуск 1189

О. С. БОРОВИК
А. В. ПАРФЕНОВ
В. И. СЫРЯМКИН

УВЛЕКАТЕЛЬНЫЕ ИГРЫ НА БЫТОВОМ КОМПЬЮТЕРЕ



МГП «Раско»
Томск 1992 г.

ББК 73
Б 83
УДК 681 322

Боровик О.С., Парфенов А.В., Сырямкин В.И.

Б 83 Увлекательные игры на бытовом компьютере. Под ред. В.И. Сырямкина – Томск: МГП "РАСКО", 1992, 107 с.: ил. – (Массовая радиобиблиотека. Вып 1189).

ISBN 5-256-00987-7

Рассмотрена структура персональных ЭВМ. Проанализированы характеристики домашних ПЭВМ отечественного и зарубежного производства. Дана классификация компьютерных игр. Описано аппаратное и программное обеспечение бытовой ПЭВМ "Спектрум". Приведены рекомендации по эксплуатации микрокомпьютера (подключение к телевизору, магнитофону, принтеру, а также вопросы, связанные с настройкой и расширением функциональных возможностей ПЭВМ). Дано описание некоторых игр и игровых программ.

Для подготовленных радиолюбителей специалистов по вычислительной технике, для студентов вузов и школьников.

Б $\frac{2404040000-036}{046(01)-92}$ КБ-11-137-92 ББК 73

Рецензент: канд. техн. наук В.С. Шидловский

ISBN 5-256-00987-7

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Область применения компьютеров достаточно широка. Они используются в быту для обеспечения интересного досуга. Одной из главных причин использования персональных ЭВМ (ПЭВМ) в быту являются увлекательные компьютерные игры, которыми можно обеспечить любой домашний компьютер.

Разработкой компьютерных игр (программным обеспечением) занимаются различные организации как в СССР, так и за рубежом. Однако чтобы реализовать компьютерные игры, необходимо иметь бытовую ПЭВМ.

Бытовые ПЭВМ представляют собой широкий спектр отечественного и зарубежного производства. В книге описывается зарубежный бытовой компьютер семейства "Спектрум" (Англия), имевший массовое применение за рубежом в 80-е годы. Этот компьютер можно было легко купить или заказать по почте в США, Японии и странах Западной Европы. Компьютер завоевал популярность за счет простоты, низкой стоимости, возможности использования бытового цветного телевизора в качестве дисплея, а бытового магнитофона – накопителя на магнитной ленте (т.е. устройства для долговременного хранения программ). Наличие в этом компьютере развитой цветной графики сделало компьютерные игры красочными и увлекательными.

Различные модификации ПЭВМ "Спектрум" получили в настоящее время широкое распространение и в СССР. Этот компьютер изготавливают радиолюбители и промышленные предприятия (например, производственное объединение "Ротор", г. Черкассы). Перспективность распространения этой ПЭВМ определяется также тем, что к ней разработано развитое программное обеспечение (включая компьютерные игры).

Однако освоение микрокомпьютеров "Спектрум" идет медленно из-за отсутствия различных обобщающих книг, содержащих справочный, описательный и другой полезный материал. В книге сделана попытка восполнить указанный пробел в научно-технической и популярной литературе.

Книга содержит два раздела и приложения.

В первом разделе дается краткий исторический обзор по созданию ЭВМ, рассматривается структура ПЭВМ, объясняется назначение ее узлов, описываются бытовые ПЭВМ, приводятся характеристики отечественных и зарубежных бытовых ПЭВМ, дается классификация компьютерных игр.

Второй раздел посвящен описанию ПЭВМ "Спектрум". Вначале приводится история создания микрокомпьютера, а затем его аппаратная реализация и программное обеспечение. Подробно рассматривается структурная схема ПЭВМ и основного центрального

процессора – микропроцессора Z80. Большое внимание уделено программированию на языке Бейсик применительно к ПЭВМ "Спектрум". Здесь особый интерес имеет программное обеспечение для отображения цветной информации и цветной графики на телевизоре. Описывается также клавиатура и режимы ее работы. Приводится материал по эксплуатации ПЭВМ "Спектрум" (подключение компьютера к телевизору, магнитофону, принтеру; рекомендации по защите программ). Дается описание некоторых компьютерных игр ПЭВМ "Спектрум".

В приложении описывается программатор ПЗУ, используемый в ПЭВМ "Спектрум". В приложении также приведены некоторые игровые программы на языке Бейсик для ПЭВМ "Спектрум".

Разделы 2.2, 2.3.1, 2.3.2, 2.4 написал О.С. Боровик; разделы 1, 2.1, 2.3.3, приложения 1, 2 – В.И. Сырямкин; 1.4 – В.И. Сырямкин совместно с А.В. Парфеновым; 2.5 – О.С. Боровик совместно с А.В. Парфеновым.

Авторы сознают, что книга не лишена недостатков, и с благодарностью примут любую критику читателей. Все замечания и предложения по книге просим направлять по адресу: 634055, Томск, а/я 2211, МГП "РАСКО"

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

АЦП	– аналого-цифровой преобразователь
АМ	– акустический модем
БИС	– большая интегральная схема
ГД	– графический дисплей
ГК	– графический контроллер
ГП	– графопостроитель
Д	– дисплей, дешифратор
Кбайт	– килобайт
КМ	– кассетный магнитофон
К	– курсор
МИС	– малая интегральная схема
НГМД	– накопитель на гибком магнитном диске.
НМД	– накопитель на магнитном диске
ОЗУ	– оперативное запоминающее устройство
НМЛ	– накопитель на магнитной ленте
П	– принтер
ПДП	– прямой доступ к памяти
ПЗУ	– постоянное запоминающее устройство
ПИА	– программируемый интерфейсный адаптер
ПЭВМ	– персональная ЭВМ
Р	– регистр
РПЗУ	– репрограммируемое ПЗУ
СИС	– средняя интегральная схема
Т	– таймер
ТД	– телевизор, дисплей
ТК	– телевизионная камера
УВЦИ	– устройство ввода цветных изображений
ЦАП	– цифроаналоговый преобразователь
ЦП	– центральный процессор
ША	– шина адреса
ШД	– шина данных
ЭВМ	– электронная вычислительная машина
ЭЛТ	– электронно-лучевая трубка

1. ОБЩИЕ СВЕДЕНИЯ О ПЕРСОНАЛЬНЫХ ЭВМ

1.1. Краткий исторический обзор создания ЭВМ

Известно, что первые ЭВМ разрабатывались в период второй мировой войны специально для военных целей [1]. Создание ЭВМ стало возможным за счет успехов в технологии производства быстродействующих цифровых (импульсных) схем, применяемых в радиолокации. После войны, во второй половине 40-х годов, при проектировании ЭВМ преследовались главным образом научные и коммерческие интересы

В начале 50-х годов появились первые ЭВМ общего назначения, в которых для построения основных логических схем (вентилей и триггеров) в качестве активных элементов использовались электровакуумные лампы. Кроме того, из-за значительного количества выделяемого тепла машины нуждались в системе охлаждения. Поэтому первые ЭВМ дороги в производстве и эксплуатации. Программа в таких ЭВМ набиралась на коммутационной панели с помощью проводников. Кроме данных, необходимых для выполнения программы, в памяти ЭВМ не содержалось никакой информации. Дальнейшим этапом совершенствования ЭВМ стал реализованный принцип запоминаемой программы, явившийся фундаментальной концепцией развития ЭВМ.

Большое влияние на развитие ЭВМ оказала замена электровакуумных приборов полупроводниковыми элементами (транзисторами, диодами и др.), что позволило значительно уменьшить вес, габариты, тепловое излучение и повысить надежность работы ЭВМ.

В начале 60-х годов совершенствование ЭВМ на базе полупроводников происходило по следующим двум направлениям: создание больших ЭВМ и создание малых (мини) ЭВМ. Первые ЭВМ, обрабатывающие большие массивы информации, были дорогостоящи, громоздки. Для их эксплуатации требовались большие помещения с кондиционерами и значительный обслуживающий персонал. Чтобы окупить большие расходы, связанные с приобретением ЭВМ, необходимо было пользоваться ею круглосуточно в течение многих недель и месяцев. По сравнению с большими ЭВМ мини-ЭВМ обладали такими преимуществами, как малые стоимость, вес и габариты. Была получена возможность работать на той ЭВМ, которая ориентирована на решение конкретного класса задач.

Дальнейшее совершенствование ЭВМ происходило одновременно с развитием технологии производства полупроводниковых схем. Появление малых, средних и больших интегральных схем (МИС, СИС, БИС) обеспечило создание персональных ЭВМ, которые уже могли располагаться на рабочем столе пользователя [2]. В середине 70-х годов с появлением микропроцессоров (БИС, реализующих архитектуру ЭВМ в одном корпусе) были созданы карманные ЭВМ (калькуляторы). Кроме того, широкое использование БИС при реализации ЭВМ позволило создать мощные универсальные машины. Развитие современных ЭВМ идет в направлении повышения быстродействия и надежности работы, снижения стоимости, веса и габаритов, а также улучшения сервиса.

1.2. Структура ПЭВМ

Обобщенная структурная схема вычислительной системы на основе ПЭВМ или настольных микроЭВМ дана на рис. 1.1. Разъясним назначение ее узлов [2].

Шина данных и адресов предназначена для электрической связи узлов вычислительной системы. Через эту шину, представляющую собой обычный или плоский многожильный кабель, ЭВМ обменивается с другими узлами цифровыми данными и адресами по которым эти данные должны поступать.

Центральный процессор (ЦП) выполняет основные вычислительные (логические и арифметические) операции. Обычно в состав ЦП входят арифметико-логическое устройство, устройство управления и устройство связи ЦП с периферийным оборудованием через общую шину.

Клавиатура (клавишный пульт) обеспечивает ввод данных и программ в ПЭВМ.

Интерфейс клавиатуры представляет собой аппаратно-программные средства сопряжения ЦП с клавиатурой.

Таймер формирует временные последовательности (сигналы), задающие порядок работы ПЭВМ.

Оперативное запоминающее устройство (ОЗУ) обеспечивает хранение и временное (оперативное) запоминание цифровой информации. ОЗУ ПЭВМ обычно включает в себя ОЗУ дисплея, необходимое для хранения одной или нескольких страниц информации.

Постоянное запоминающее устройство (ПЗУ) осуществляет постоянное хранение микропрограмм, предназначенных для выполнения основных операций ПЭВМ на машинно-ориентированном языке. ПЗУ содержит также программу-интерпретатор (или транслятор) для программирования ПЭВМ на языке высокого уровня например Бейсик, Квейсик, Си и др.

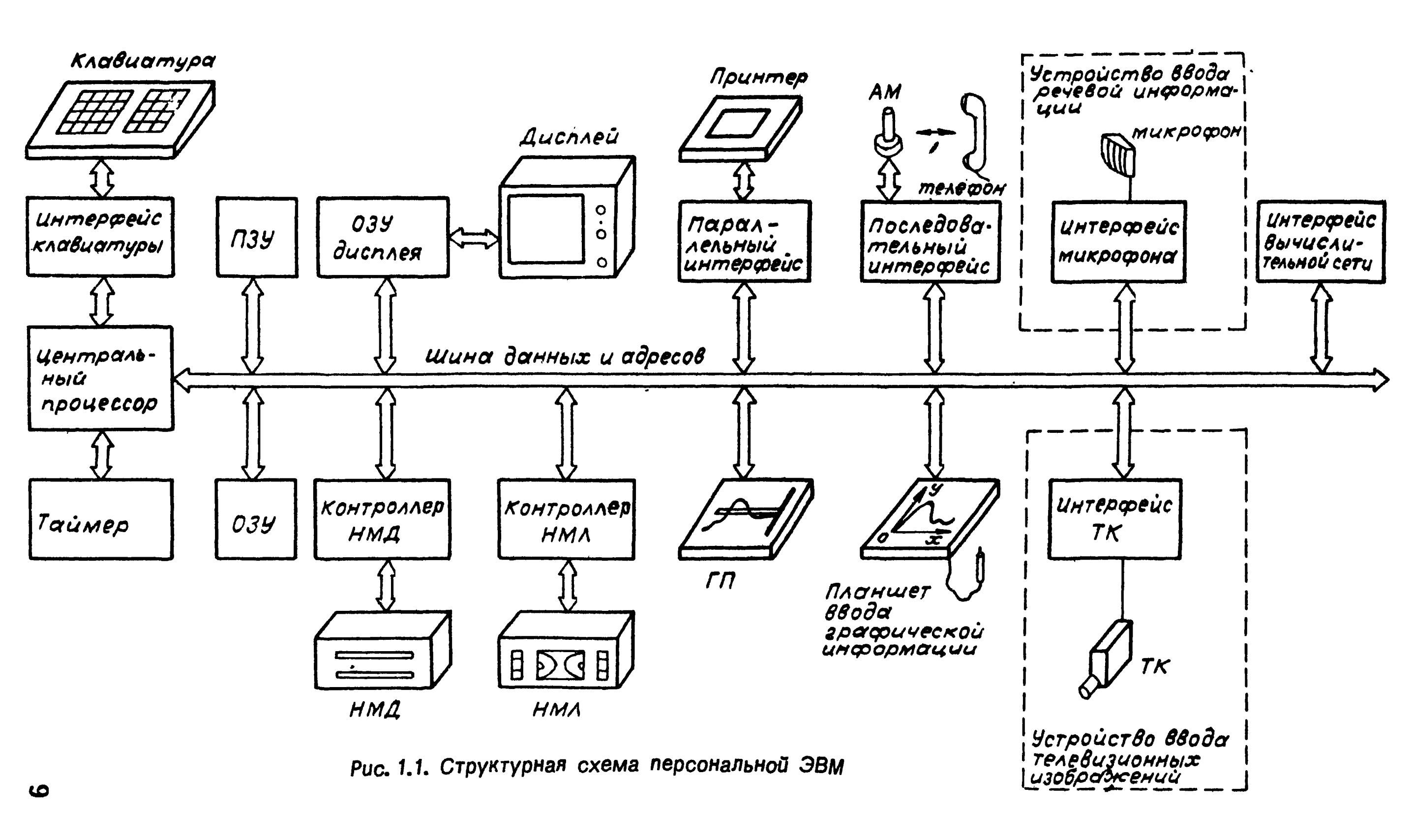


Рис. 1.1. Структурная схема персональной ЭВМ

Накопитель на магнитном диске (НМД) представляет собой устройство для магнитной записи цифровой информации на носитель в виде диска с ферромагнитным покрытием и считывания информации с диска. НМД, дополняющий ПЗУ ПЭВМ, может хранить информацию, вводимую пользователем (массивы данных и программы).

Накопитель на магнитной ленте (НМЛ), выполняющий функции, аналогичные функциям НМД, представляет собой бытовой или специализированный кассетный магнитофон для записи цифровой информации на магнитную ленту.

Контроллер НМД и контроллер НМЛ осуществляют сопряжение ЦП с соответствующими накопителями.

Принтер производит распечатку алфавитно-цифровой (иногда и графической) информации.

Параллельный интерфейс представляет собой аппаратно-программные средства сопряжения ПЭВМ с периферийным оборудованием, воспринимающим информацию в виде параллельного кода (например, принтером).

Дисплей предназначен для визуального отображения алфавитно-цифровой и графической информации (вводимой, выводимой, промежуточной).

Акустический модем (АМ), используемый для связи ПЭВМ с вычислительными центрами, осуществляет преобразование информации в высокочастотные звуковые сигналы, посылаемые по телефонной сети.

Последовательный интерфейс представляет собой аппаратно-программные средства сопряжения ПЭВМ с периферийным оборудованием, воспринимающим информацию в виде последовательного кода (например, АМ).

Графопостроитель (ГП) производит построение графиков и рисунков с высокой разрешающей способностью.

Планшет для ввода графической информации обеспечивает преобразование координат точек графика в электрические сигналы, вводимые в ПЭВМ.

Устройство ввода телевизионных изображений содержит телевизионную камеру (ТК) и соответствующий интерфейс связи ТК с ПЭВМ, обеспечивает ввод–вывод видеоинформации в микроЭВМ [3].

Устройство ввода речевой информации включает микрофон и соответствующий интерфейс связи микрофона с ПЭВМ, осуществляющий ввод звуковой (речевой) информации в ЭВМ.

Интерфейс вычислительной сети выполняет связь между ПЭВМ и другими ЭВМ. С помощью этого интерфейса можно создавать вычислительные сети различной конфигурации.

Следует отметить, что перечисленные устройства полностью могут входить только в наиболее дорогие и сложные ПЭВМ. В зависимости от реализации и комплектации ПЭВМ имеют

различные характеристики. К основным характеристикам ПЭВМ относятся: объем памяти ОЗУ и ПЗУ, скорость счета, языки программирования, габариты и масса, состав периферийного оборудования.

1.3. Типы ПЭВМ и их возможности

Персональные ЭВМ, предназначенные для использования одним пользователем, обеспечивают децентрализацию вычислительных процессов, резкое сокращение стоимости машинного времени, повышение удобства эксплуатации и привлечение к работе на ЭВМ обширного круга пользователей – неспециалистов в области вычислительной техники. Это достигается за счет простой диалоговой работы пользователя с ПЭВМ [2].

Все персональные ЭВМ делятся на три класса. карманные, домашние, профессиональные.

Карманные ПЭВМ (калькуляторы) – это сверхминиатюрные ЭВМ, выполненные на БИС большой степени интеграции и предназначенные для решения достаточно сложных и простых информационных задач. **Домашние (бытовые) ПЭВМ** (ПЭВМ среднего класса) ориентированы на решение различных задач (расчетных, информационных) средней сложности, в основном относящихся к быту: расчеты семейного бюджета, обучение различным дисциплинам, учебные расчеты, простые и сложные "интеллектуальные игры" (шахматы, космические игры и др.). **Профессиональные ПЭВМ** предназначены для решения различных сложных задач, выполняемых профессиональными работниками (математиками, физиками, программистами, экономистами). Профессиональные ПЭВМ могут эксплуатироваться как на работе, так и дома.

Следует отметить, что разрабатываемые ПЭВМ должны обладать следующей совокупностью свойств [2]:

- ориентацией на эксплуатацию одним пользователем (неспециалистом в области вычислительной техники);
- низкой стоимостью за рубежом от 50 до 5000 дол., в СССР от 70 до 25000 руб.), позволяющей приобретать ЭВМ небольшим организациям и даже пользователям за счет личных средств;
- простой конструкцией, высокой надежностью, малыми габаритами и весом, позволяющими размещать ЭВМ на рабочем столе;
- работой с обширным периферийным оборудованием (рис. 1.1 и § 1.4);
- программированием хотя бы на одном языке высокого уровня (Бейсик, Квейсик, Паскаль, Си и др.);
- реализацией общения с пользователем в письменной или устной форме (за счет развитых устройств ввода-вывода информации);
- обеспечением возможности решения широкого круга задач (расчетных, вычислительных, информационных).

1.4. Домашние (бытовые) ПЭВМ

Бытовые ПЭВМ обычно рассчитаны на подключение к кассетному магнитофону и телевизору (или дисплею). Эти ЭВМ изготавливаются в виде небольшой клавиатуры. Такое конструктивное решение обеспечивает резкое удешевление ПЭВМ и удобство ее эксплуатации [2]. Характеристики ряда простейших зарубежных ПЭВМ даны в табл. 1.1.

Таблица 1.1

Характеристика простейших зарубежных ПЭВМ

Характеристика	Тип ЭВМ			
	ZX-80	ZX-81	HP-75	PB-700
Фирма и страна-изготовитель	Sinclair Radions Ltd (Великобритания	Sinclair Radions Ltd (Великобритания	Hewlett Packard (США)	Casio (Япония)
Разрядность микропроцессора	8	8	8	8
Емкость ПЗУ кбайт	32	32	48 + 16	25
Емкость ОЗУ, кбайт	1	2	16 + 8	4 + 16
Число строк/знаков дисплея (телевизора)	Внешний 24/32	Внешний 24/32	1/32	4/20
Габариты, мм	299x178x x51	299x178x x51	–	200x85x x20
Масса, кг	0,34	0,34	0,75	-
Периферийные устройства	ТД, П, НМЛ	ТД, П, НМЛ	НМЛ, АМ, П	НМЛ, П
Язык программирования	Бейсик	Бейсик	Бейсик	Бейсик

Представленные ПЭВМ ориентированы в основном на работу с черно-белым телевизором и осуществляют формирование алфавитно-цифровых признаков и графиков. Исключение составляет ПЭВМ HP-75, которая содержит встроенный дисплей на одну строку и накопитель информации на магнитных картах.

Характеристики некоторых отечественных бытовых ПЭВМ даны в табл. 1.2. Как видно из таблицы, самой простейшей ПЭВМ является ПЭВМ "Специалист" [4]. Сейчас широко используется ПЭВМ

Таблица 1.2

Характеристики некоторых отечественных бытовых ПЭВМ

Характеристика ЭВМ	Тип ЭВМ					
	"Специалист"	"Радио-86РК"	"Электроника-БК-0010"	"Партнер"	"Орион-128"	"Электроника МК90"
Разрядность микропроцессора	8	8	16	8	8	16
Емкость ПЗУ, кбайт	2 – 14	2	32	16	2	32
Емкость ОЗУ, кбайт	48	16 или 32	32+4 или 16	64	128	16
Число строк/знаков дисплея (телевизора)	25/64	25/64	24/64 или 32	25/64 или 80	25/64	64/120
Габариты, мм	160x240x30	160x260x30	370x180x70	418x334x68	-	30x250x100
Масса, кг	около 0,5	около 0,5	4,0	3,7	-	0,7
Периферийные устройства	НМЛ, П	НМЛ, П	НМЛ, П	НМЛ, П	НМЛ, П	Внешняя память. ОЗУ – 10 кбайт, ПЗУ – 192 кбайт
Язык программирования	Бейсик, Ассемблер, Паскаль	Бейсик, Ассемблер	Фокал, Бейсик, Ассемблер	Бейсик, Ассемблер, Форт	Бейсик, Ассемблер	Бейсик
Стоимость, руб.	Промышленностью не выпускаются		540	650	-	3500

"Радио-86 РК" для бытовых и даже производственных целей [5]. Большими возможностями обладает бытовая ПЭВМ "Электроника-БК-0010" [2, 6, 7]. Она предназначена для выполнения расчетов, создания каталогов, справочников, словарей, учебных программ и, конечно, предоставляет пользователю компьютерные игры. К данным ПЭВМ "Электроника-БК-0010", представленным в табл. 1.2, следует добавить, что различие в выводе знаков в строке на дисплей определяется размерностью матрицы знаков (8x8 или 16x8), а клавиатура имеет 92 клавиши.

ПЭВМ "Партнер" состоит из центрального вычислительного модуля "Партнер-01.01" и дополнительных функциональных модулей, позволяющих пользователю оперативно формировать компьютер необходимой конфигурации. В модуле "Партнер-01.01" размещены блок питания, клавиатура и вычислитель. В качестве дисплея может использоваться бытовой телевизор или монитор МС-6501. К ПЭВМ "Партнер" можно подключать дополнительные модули, предназначенные для расширения функциональных возможностей компьютера. Эти модули следующие: расширение ОЗУ и ПЗУ; программатор ПЗУ; дополнительный видеоконтроллер (черно-белая и цветная графика); музыкальный синтезатор; игровые манипуляторы; измерительные и исполнительные устройства для лаборатории (мультиметр, осциллограф и др.) [8].

Большими возможностями обладает ПЭВМ "Орион-128", собранная, как компьютеры "Специалист", "Радио-86 РК" и "Партнер", на микропроцессоре КР580ВМ80А. Бытовая ПЭВМ "Орион-128" проста, так как реализована на 8-разрядном процессоре. Комплект деталей этой ПЭВМ не содержит остродефицитных компонентов, и их стоимость мала. Следует отметить, что для 8-разрядных ПЭВМ накоплено большое количество программ (системных, прикладных, игровых). Это значительно облегчает адаптацию программ к ПЭВМ "Орион-128". Компьютер "Орион-128" обладает большими графическими и цветовыми возможностями (обработка 16 цветов при разрешающей способности 256x256). ПЭВМ "Орион-128" обеспечена максимальной преемственностью с компьютером "Радио-86РК", что значительно помогает радиолюбителям, уже имеющим эту бытовую ПЭВМ, заниматься реализацией новой ПЭВМ. Такая преемственность достигнута благодаря общности в структуре программы МОНИТОР, идентичному для обеих ПЭВМ стандарту ввода-вывода на магнитофон, применению в ПЭВМ "Орион-128" клавиатуры от ПЭВМ "Радио-86РК". Кроме того, в базовый вариант ПЭВМ "Орион-128" заложено расширение возможностей компьютера за счет следующих особенностей: открытой архитектуры, буферизирования шины данных, адреса и управления, наличия зарезервированных выходов для управления дополнительными блоками памяти и портами ввода-вывода, входа блокирования основного ОЗУ и входов RDI ("готовность"), INT ("запрос прерывания") микропроцессора [9].

Промежуточное положение между бытовыми и персональными ПЭВМ занимает "Электроника МК90", имеющая жидкокристаллический графический дисплей и сменные модули внешней памяти (ОЗУ, ПЗУ) [10]. Хотя эта ПЭВМ имеет малые габариты и вес, но значительная стоимость пока не позволяет широко использовать ее в быту.

Следует отметить, что такие "портфельные" ПЭВМ сейчас начинают широко использоваться у нас в стране и за рубежом.

Новое поколение ПЭВМ этого класса рассчитано на сопряжение с цветным дисплеем или телевизором (табл. 1.3) [2]. Они способны обрабатывать (воспроизводить) многоцветные полутоновые изображения, строить графики и рисунки с высокой разрешающей способностью и различными цветовыми оттенками Как видно из табл. 1.3, ПЭВМ имеют довольно незначительные отличия друг от друга. Все ПЭВМ этого класса обеспечивают более высокие возможности (по сравнению с простейшими вариантами ПЭВМ представленными в табл. 1.1) в решении различных информационных задач (игры, просмотр компьютерных слайдов и т.п.). В то же время в решении математических задач эти ПЭВМ (из-за ограниченности математических операторов) могут уступать простейшим компьютерам.

Одной из наиболее распространенных моделей нового поколения ПЭВМ является микроЭВМ "ZX-Spectrum" английской фирмы "Sinclair Radions Ltd" [2, 11, 12]. Клавишный пульт "ZX-Spectrum" (размером 233x144x30 мм) обеспечивает ввод большинства команд расширенной версии Бейсика нажатием одной-двух клавиш. Поэтому общее число клавиш ПЭВМ снижено до 40 (за счет использования префиксных клавиш). Кроме стандартных команд Бейсика клавиатура осуществляет ввод до 37 графических символов из 22 кодов управления цветом. В клавиатуру вмонтирован звуковой синтезатор, воспроизводящий до 130 полутонов звука. Емкость ПЗУ – 16 кбайт. При отображении графики в 8 цветах матрица экрана содержит 256x176 точек, а при отображении алфавитно-цифровой информации на экране формируется 25 строк с 32 символами в каждой. Для этой ПЭВМ специально разработаны малогабаритные принтер и накопитель на диске. ПЭВМ может подключаться к кассетному магнитофону и обеспечивает скорость записи до 1500 бит/с (например, данные или программы, занимающие ОЗУ 16 кбайт, записываются (и вводятся в ПЭВМ) за время 100 с). В модифицированном варианте ПЭВМ "ZX-Spectrum" число клавиш расширено до 58 (это позволило использовать новую расширенную версию языка Бейсик и реализовать ряд дополнительных текстовых и графических операторов), емкость ОЗУ увеличена до 41,5 кбайт. Накопитель на кассете с бесконечным рулоном ленты обеспечивает емкость внешнего ПЗУ до 85 кбайт на одну кассету. Загрузка программ емкостью 48 кбайт обеспечивается за 9 с [2, 13].

Таблица 1.3

Характеристики некоторых зарубежных ПЭВМ, сопрягаемых с цветным дисплеем (телевизором)

Характеристика	Тип ЭВМ					
	ZX-Spectrum	BBC-Micro A	VIC-20	ATARI-400	TI-99/4A	TRS-Colour
Емкость ОЗУ, кбайт, стандартная/максимальная	16/48	16/32	5/29	16/16	16/48	16/32
Число цветов (общее)	8	8	16	16	16	9
Емкость ОЗУ, кбайт (для дисплея)	9	3	–	7	14	10
Число цветов, выводимых одновременно	8	4	16	4	16	8
Режим мерцания	Есть	Есть	Нет	Нет	Нет	Нет
Подстройка яркости	Есть	Нет	Нет	Есть	Нет	Нет
Установка графического режима пользователем	Есть	Есть	Есть	Нет	Есть	Нет
Полный набор кода ASCII	Есть	Есть	Нет	Нет	Нет	Нет
Экран дисплея (строки/столбцы)	24/32	25/40	23/24	24/40	24/29	16/32
Автоповтор на всех клавишах	Есть	Есть	Нет	Есть	Есть	Нет
Графика повышенной разрешающей способности	Есть	Есть	Нет	Есть	Нет	Есть
Язык программирования	Бейсик,	Бейсик	Бейсик	Бейсик	Бейсик	Бейсик

1.5. Компьютерные игры

Компьютерные игры, осуществляемые в диалоговом режиме с ПЭВМ, делятся на:

- обучающие игры;
- развивающие или лечебные игры;
- развлекательные игры;
- сложные и "интеллектуальные" игры.

К обучающим играм относятся программы ПЭВМ, обеспечивающие изучение каких-то правил или учебных дисциплин. Например, программы, обучающие правилам и навыкам вождения (программы-имитаторы мотогонок, велогонок, автогонок, гонок на моторных лодках, тренажеры по отработке навыков вождения легкового автомобиля), а также программы по изучению иностранных языков и других учебных предметов (физика, математика и др.). При разработке особо выделяют детские обучающие программы, которые должны удовлетворять известным критериям:

- обязательно чему-то учить;
- просто осваиваться;
- не содержать внешние подзагрузочные уровни;
- приводить к результату (хотя бы и промежуточному) в течение непродолжительного времени;
- не вызывать раздражительность, эмоциональные перегрузки и стрессы.

Развивающие игры предназначены для развития каких-либо навыков. Например, игровые программы НТК "Инфорком" могут развивать навыки фотографического схватывания текста. Программа может автоматически настраиваться на играющего (после включения программа автоматически усложняется до первой ошибки играющего, а затем упростится, снова начнет плавно усложняться). Другие программы этого научно-технического кооператива служат для развития периферийного зрения и развития навыков блочного восприятия текстов. Лечебные игры способствуют восстановлению некоторых утраченных функций человека (например, восстановление реакции или двигательной способности органов движения и др.).

Развлекательные игры бывают простыми и сложными и обеспечивают неутомительный отдых (охота, морской бой, воздушный бой и т. п.).

Интересными являются сложные и "интеллектуальные" игры, к которым относятся математические задачи (логические головоломки, математические развлечения, азартные игры, зашифрованные операции [14], игра в шахматы, посадка космических кораблей на планеты и др.). Отметим, что наиболее сложными являются логические головоломки трех основных типов: головоломки, которые интересно программировать; задачи, имеющие очень простой вид,

но доведение их решения до программы является настоящей головоломкой; программа, по которой необходимо определить исходное математическое выражение [14]

2. БЫТОВАЯ ПЭВМ "СПЕКТРУМ"

2.1. История создания ПЭВМ "Спектрум"

Изобретатель ПЭВМ "Спектрум" ("ZX-Spectrum") Клайв Марлз Синклер [11, 12, 15] в 1961 году в Англии зарегистрировал свою первую компанию "Синклер Радионикс", выпустившую в 1963 году микроусилитель. В разработках Синклер всегда ставил две сверхзадачи – минимальные габариты и минимальную цену. При этом он стремился быть первым на рынке со своим товаром.

Первый значительный успех пришел к Синклеру в 1979 году. В это время фирма "Коммодор" выпустила впервые бытовой компьютер "PET" стоимостью 700 ф. ст. В этой связи газета "Файнэншл Таймс" прогнозировала понижение цены на персональный компьютер до 10 ф. ст. не ранее чем через 5 лет. Однако Синклер уже через полгода выпустил микрокомпьютер ZX-80 ценой 99 ф. ст. Такое резкое снижение цены ПЭВМ обусловлено возможностью использования в качестве дисплея бытового телевизора, а в качестве внешнего накопителя – бытового магнитофона [15].

Этот компьютер имел большой успех на рынке: в первые 8 месяцев было продано 20 тыс. компьютеров. Затем в марте 1981 года была выпущена новая модель ПЭВМ ZX-81 стоимостью 69 ф. ст., а через несколько месяцев – принтер к нему.

Компания Синклера в это время ведет интенсивную торговлю через различные зарубежные фирмы: "Таймекс" (США), "Митцуи" (Япония) и британскую книготорговую сеть. Только за один год товарооборот компании достиг 30 млн. ф. ст.

Несмотря на это Синклер в марте 1982 года разработал две версии новой модели "Спектрума" ("Спектрум-16", "Спектрум-48"), популярность которой превзошла все ожидания [16]. Задумывался этот микрокомпьютер как учебный для изучения программирования. Но фирмы, разрабатывающие программное обеспечение, использовав хорошую динамичную графику ПЭВМ, стали выпускать увлекательные компьютерные видеоигры. Эти компьютеры имели большой успех и продавались по 15 тыс. штук в неделю. И хотя к 1984 году фирмы "Атари", "Коммодор" и "Амстрад" выпустили компьютеры, превосходящие ПЭВМ "Спектрум", рынок был смещен в сторону изделий Синклера. Они уже выпускались более чем в 30 странах мира [15, 17–19].

В 1984–1985 годах компания Синклера выпустила две усовершенствованные модели ПЭВМ: "Спектрум+", снабженную усовер-

шенствованной клавиатурой, и "Спектрум +128" ("Дерби"), имеющую ОЗУ и ПЗУ с соответствующими объемами памяти 128 и 32 кбайт

И все-таки инженер Синклер так и не стал бизнесменом. Под давлением финансовых и организационных трудностей, связанных с неоправдавшей себя экономически 32-разрядной моделью "Sinclair-QL", в 1986 году компания "Sinclair Research Limited" была вынуждена продать все права на производство компьютера "Спектрум" французской фирме "Амстрад". Это было вызвано тем, что указанная ПЭВМ планировалась как дешевая альтернатива американским компьютерам IBM PC , но в ее концепции было заложено несколько просчетов (например, использование в качестве внешней памяти микродрайвов вместо дисководов). Поэтому эта модель ПЭВМ Синклера не стала бытовой по цене и профессиональной – по своей реализации [15].

Поскольку последующие модели ПЭВМ – "ZX-Spectrum+2" (1986) со встроенным магнитофоном; "ZX-Spectrum+3" (1987) со встроенным дисководом – выпускались уже фирмой "Амстрад", то они были похожи по внешнему виду на компьютер "Амстрад-6128". Эти компьютеры имели полноценную клавиатуру, но в то же время встроенные магнитофон и дисковод значительно увеличивали цену ПЭВМ. К тому же ПЭВМ создавали неудобства пользователю из-за нестандартного диаметра дискет 3,0 дюйма, малой их емкости (180 кбайт) и практической сложности переноса имеющихся кассетных версий программ на диск.

С 1986 года фирма "Таймекс" (на заводах в Португалии) начала выпускать компьютеры, совместимые с компьютером "Спектрум". "Таймекс-2048" – для поставки в Европу и "Таймекс-2068" – для поставки в США. ПЭВМ "Таймекс-2048" имела улучшенную клавиатуру, встроенный порт манипулятора "джойстик", светоиндикатор и выключатель питания, две экранные области памяти, режим расширенной цветной графики. ПЭВМ "Таймекс-2068" включала сменные картриджи (при поочередном подключении этих устройств компьютер становился либо Спектрум-совместимым, либо CP/M-совместимым).

Отметим также, что планировалась к выпуску в 1987 году перспективная модель "Суперспектрум" ("Локки"), в которую был заложен процессор Z80H, имеющий частоту 7 мГц. ПЭВМ позволяла обслужить два банка памяти по 64 кбайт и экран емкостью более 51 кбайт, кроме этого, машина имела разрешающую способность 192x256 пикселов с возможностью одновременного воспроизведения 64 цветов (для каждой точки). "Локки" была программно совместима со "Спектрумом", стоила менее 200 ф.ст. и представляла серьезную конкуренцию ПЭВМ семейства "Амиги". Однако фирма "Амстрад", пользуясь своими правами, опасаясь конкуренции для своих ПЭВМ, отказала в разрешении на его производство [15].

2.2. Описание компьютера

2.2.1. Структурная схема компьютера

Существует множество вариантов реализации ПЭВМ "Спектрум", получивших распространение у нас в стране. Печатные платы и комплектующие для них можно приобрести в отделах "Радиотовары", у кооператоров или выписать по почте. Начат выпуск нескольких серийных моделей "Магик", "Компаньон", "Робик" и др. И хотя все они различаются элементной базой, схемотехническими решениями и конструктивно, принцип работы у них один.

Ниже приводятся обобщенная структурная схема компьютера и назначение функциональных блоков, входящих в его состав (рис. 2.1).

Согласно структурной схеме ПЭВМ "Спектрум" состоит из системного генератора и трех частей: системной ("SYS"), графической ("GRAF") и части, обеспечивающей обмен информацией с устройствами ввода вывода "IIO" (на рис. 2.1 выделены пунктирной линией).

Системный генератор формирует тактирующие сигналы для центрального процессора и формирователя телевизионных сигналов, а также тактовую последовательность импульсов для графического контроллера и формирователя адресов (часть "GRAF").

К системной части "SYS" относятся следующие устройства.

Центральный процессор (ЦП) Z80. Адресные линии (ША) ЦП связаны с адресными входами системного контроллера, системного РПЗУ, коммутатора адресов (строки столбцы) системного ОЗУ, регистрами адреса строк и столбцов (часть "GRAF") и дешифратором адресов портов клавиатуры (часть "IIO").

Шина данных (ШД) ЦП не буферизирована и непосредственно соединена с выходом данных системного РПЗУ, с входами и выходами данных системного ОЗУ, с входами данных порта вывода на телевизор (TV) и регистра данных (часть "GRAF"), с выходами данных порта вывода на магнитофон и громкоговоритель (часть "IIO"), с выходами данных порта ввода с магнитофона и клавиатуры (часть "IIO"), с выходами порта ввода с джойстика (часть "IIO").

Сигналы управления $\overline{MREQ}$ (запрос памяти) и $\overline{IORQ}$ (запрос ввода/вывода) поступают на системный контроллер, а сигналы $\overline{WR}$ (запись) и $\overline{RD}$ (чтение) – на системный контроллер и системное ОЗУ.

Более подробно микропроцессор Z80 рассматривается в 2.2.2.

Системный контроллер (СК). В зависимости от адреса, установленного ЦП на ША, а также сигналов $\overline{WR}$, $\overline{RD}$, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RESET}$ (сброс) СК вырабатывает следующие сигналы для управления всей

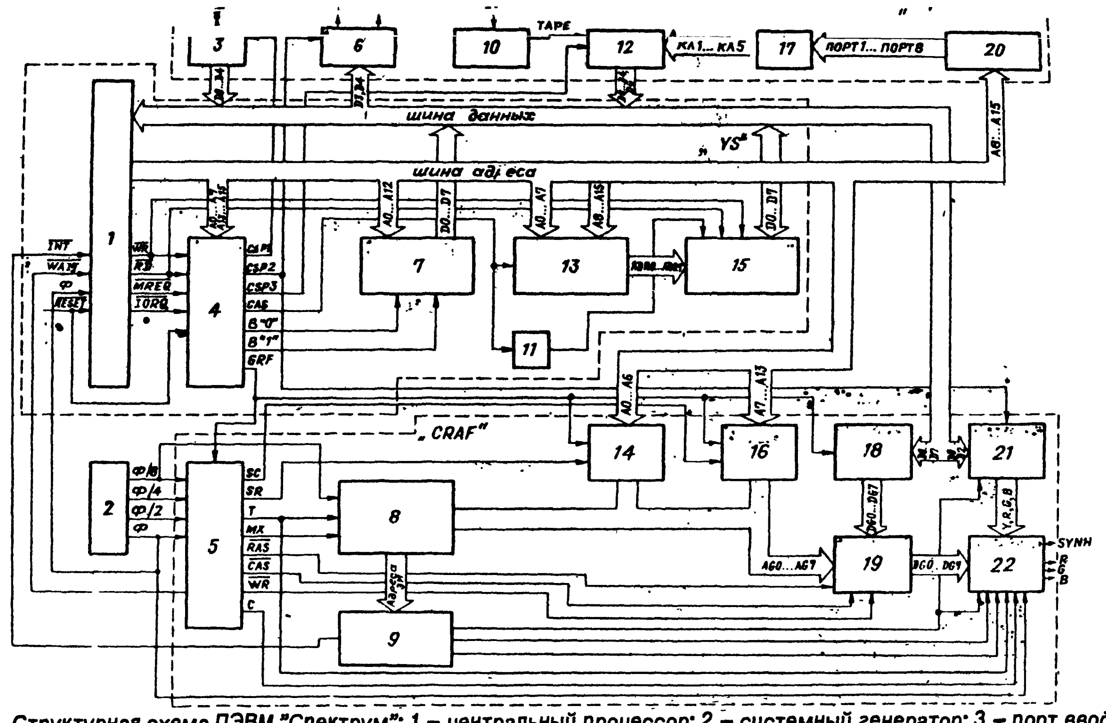


Рис. 2.1. Структурная схема ПЭВМ "Спектрум": 1 – центральный процессор; 2 – системный генератор; 3 – порт ввода с джойстика; 4 – системный контроллер; 5 – графический контроллер; 6 – порт вывода на магнитофон и громкоговоритель; 7 – системное РПЗУ (8 Кбайт×2); 8 – формирователь адресов (графика); 9 – TV–синхронизатор; 10 – формирователь сигналов с магнитофона; 11 – линия задержки; 12 – порт ввода с магнитофона и клавиатуры; 13 – коммутатор адресов строки/столбцы; 14 – регистр адреса строк (графика); 15 – системное ОЗУ (64 Кбайт); 16 – регистр адреса столбцов (графика); 17 – клавиатура; 18 – регистр данных (графика); 19 – графическое ОЗУ (16 Кбайт); 20 – дешифратор адресов портов клавиатуры; 21 – порт вывода на TV (бордюр); 22 – формирователь TV-сигналов

системой (компьютером) в целом:

сигнал CSP1 – ввод информации с джойстика;

сигнал CSP2 – вывод информации на магнитофон громкоговоритель и телевизор;

сигнал CSP3 – ввод информации с клавиатуры и магнитофона;

сигнал CAS – чтение системного ОЗУ;

сигнал B0 – чтение банка B0 системного РПЗУ;

сигнал B1 – чтение банка B1 системного РПЗУ;

сигнал GRF –обращение к "экрану".

Системное РПЗУ (8 кбайт × 2). Здесь "зашита" операционная система (ОС) "СПЕКТРУМ" – монитор и интерпретатор Бейсик.

Коммутатор адресов строки столбцы. Мультиплексирует адреса для системного ОЗУ.

Системное ОЗУ. Организация памяти компьютера рассматривается в 2.3.1.

К графической части "GRAF" относятся следующие устройства.

Графический контроллер (ГК). В зависимости от актовых последовательностей импульсов, поступающих из системного генератора, а также сигнала GRF, вырабатываемого системным контроллером, ГК формирует сигналы для управления графической частью компьютера и сигнал $\overline{\text{WAIT}}$ (ожидание), который переводит ЦП в состояние ожидания на время записи информации в графическое ОЗУ. ГК вырабатывает следующие сигналы:

сигнал SC – занесение адреса в регистр адреса столбцов;

сигнал SR – занесение адреса в регистр адреса строк;

сигналы $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, и $\overline{\text{WR}}$ – управление графическим ОЗУ;

сигнал MX – мультиплексирование адресов строк и столбцов, вырабатываемых формирователем адресов;

сигнал C – управление формирователем TV-сигналов;

сигнал T – управление формирователем TV-сигналов и формирователем адресов.

Формирователь адресов. Вырабатывает адреса строк и столбцов для графического ОЗУ и адреса элементов изображения (ЭИ) для TV-синхрогенератора.

Синхрогенератор. Вырабатывает синхроимпульсы для управления формирователем TV-сигналов и сигнал $\overline{\text{INT}}$ (запрос на маскируемое прерывание), по которому каждые 20 мс (с частотой кадровой развертки) ЦП переходит к выполнению программы обслуживания клавиатуры, инкрементирует часы и т.д.

Регистр адреса строк. Служит для хранения адреса строки графического ОЗУ.

Регистр адреса столбцов. Предназначен для хранения адреса столбца графического ОЗУ.

Регистр данных. Служит для хранения данных графического ОЗУ.

Графическое ОЗУ. Предназначено для хранения графической

информации (экрана). Экран компьютера содержит 768 знакомест (позиций) – 24 строки по 32 знакоместа. Знакоместо содержит 8x8 точек (ЭИ, или пикселов). Таким образом, экран содержит 192x255 точек и занимает адресное пространство с 16384 по 22527 (по каждому адресу хранится 8 точек в черно-белом виде). Адреса с 22528 по 23295 предназначены для атрибутов (цвет, яркость и мерцание), которые задаются для всего знакоместа, а не для отдельной точки (пиксела). Биты с 0-го по 2-й байт атрибутов задают цвет чернил, с 3-го по 5-й – цвет бумаги, 6-й бит – яркость, 7-й – признак мерцания. 24 строки экрана разбиты на 3 части. Строки с 0-й по 7-ю соответствуют адресам с 16384 по 18431. Строки с 8-й по 15-ю соответствуют адресам с 18432 по 20479. Строки с 16-й по 23-ю соответствуют адресам с 20480 по 22527. Атрибуты просматриваются для каждой строки экрана сверху–вниз, слева–направо Полностью один экран занимает 6912 байтов

Порт вывода на TV. Служит для хранения информации о цвете бордюра.

Формирователь TV-*сигналов*. Преобразовывает информацию, поступающую с графического ОЗУ и порта вывода на TV, в телевизионные сигналы "R", "G"и "B", а также выдает сигнал "SYNH" (синхросмесь) для синхронизации телевизора.

К части "IIO ", обеспечивающей обмен информацией с устройствами ввода/вывода, относятся следующие устройства.

Порт ввода с джойстика. Осуществляет ввод информации с KEMPSTON-джойстика.

Порт вывода на магнитофон и громкоговоритель. Обеспечивает вывод (запись) информации на магнитофон (бит данных D4) и громкоговоритель (бит данных D3).

Формирователь сигналов с магнитофона. Предназначен для "оцифровки" аналоговых сигналов, поступающих с магнитофона и формирования сигнала "TAPE".

Порт ввода с магнитофона и клавиатуры. Осуществляет ввод информации с магнитофона (бит данных D6) и клавиатуры (биты данных D0 ... D4).

Клавиатура. Предназначена для ручного ввода информации путем нажатия клавиш и формирования сигналов "Кл1", ... ,"Кл5", причем нажатой клавише соответствует уровень логического "0".

Дешифратор адресов портов клавиатуры В зависимости от адреса, поступающего с ША, выбирается один из восьми портов клавиатуры (клавиатура разбита на 8 рядов по 5 клавиш в каждом).

2.2.2. Микропроцессор Z80

Архитектура процессора Z80 основывается на архитектурных принципах микропроцессора 8080 фирмы "Intel" и позволяет

выполнять все 78 команд этого микропроцессора, а также 80 дополнительных команд. Всего микропроцессор Z80 имеет 696 кодов операций (в отличие от микропроцессора 8080, у которого 244 кода). К числу его особенностей относятся. использование одного источника напряжения +5 В, наличие однофазного внешнего синхрогенератора, 17 внутренних регистров и встроенная схема регенерации ОЗУ. Предусмотрены три способа реакции на прерывания. Структурная схема микропроцессора Z80 и маркировка его выводов показаны на рис. 2.2 и рис. 2.3 [20–23]

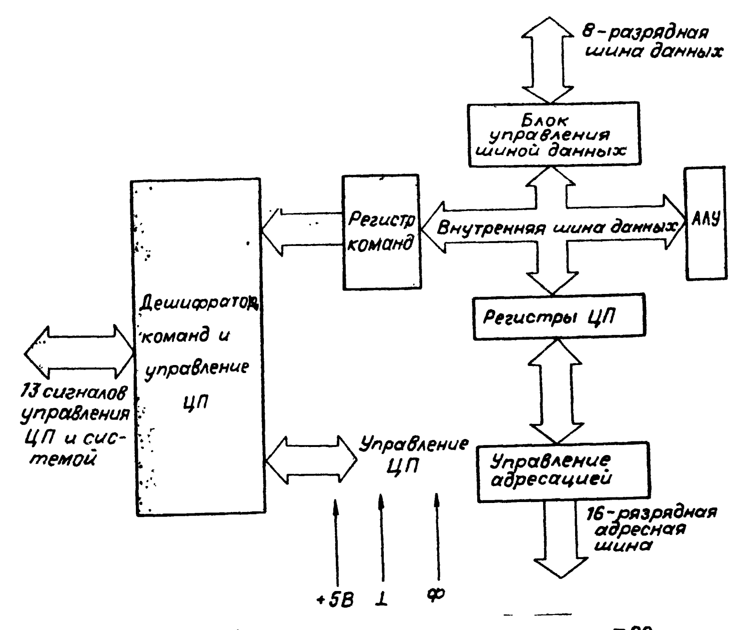


Рис. 2.2. Структурная схема микропроцессора Z 80

В микропроцессоре Z80 повторены все регистры микропроцессора 8080 и дополнительно к его 8-разрядным регистрам A, F, B, C, D, E, H и L имеются также A', F', B', C', D', E', H', и L' и еще несколько регистров специального назначения . К ним относятся два 16-разрядных индексных регистра (IX и IY), 8-разрядный регистр вектора прерываний (I) и 8-разрядный регистр регенерации памяти (R). Кроме того, имеется один 16-разрядный указатель стека.

Микропроцессор Z80 имеет два индентичных 8-разрядных флаговых регистра, в которых предусмотрены четыре проверяемых и два непроверяемых флажка. К первым четырем относятся признак переноса, признак нуля, признак отрицательного числа и признак четности – переполнения. К двум другим – признак переноса и

признак вычитания. Последний используется для коррекции результатов операций с двоично-десятичными числами, помогая определить вид предыдущей операции. Для операций сложения и вычитания коррекция различна.

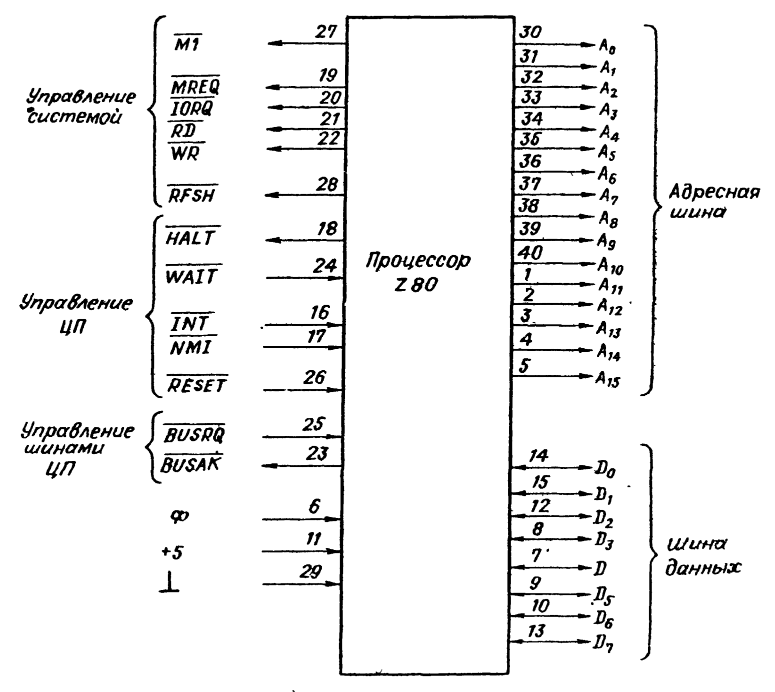


Рис. 2.3. Выводы микропроцессора Z80

Поясним теперь назначение выводов и сигналов микропроцессора Z80 [22, 23]:

АФ–А15 (шина адреса) – выходы с тремя устойчивыми состояниями. Активный уровень сигналов высокий. Используется для адресации 65535 ячеек памяти и 65535 портов ввода/вывода. В ПЭВМ "Спектрум" возможно обращение только к 32768 нечетным адресам ввода/вывода, так как при нуле на шине АФ из-за неполной дешифрации адреса всегда будет выбран 254-й порт ввода/вывода, который задействован под чтение/запись с магнитофона, вывод звукового сопровождения, вывод цвета бордюра и опрос клавиатуры. При адресации портов ввода/вывода чаще используют линии АФ–А7 (256 адресов), при этом на линиях А8–А15 присутствует содерж ое регистра В микропроцессора.

DФ–D7 (шина данных) – входы/выходы с тремя устойчивыми

состояниями. Активный уровень высокий. Используется для ввода/вывода данных микропроцессора, памяти и периферийных устройств.

$\overline{\text{M1}}$ – выход микропроцессора. Активный сигнал низкий. Указывает, что в текущем цикле осуществляется выборка кода следующей операции из памяти, а совместно с сигналом $\overline{\text{IORQ}}$ служит для подтверждения прерывания.

$\overline{\text{MREQ}}$ (запрос памяти) – выход с тремя устойчивыми состояниями. Активный уровень низкий. Указывает, что на адресной шине установлен адрес памяти, а не устройства ввода/вывода.

$\overline{\text{IORQ}}$ (запрос ввода/вывода) – выход с тремя устойчивыми состояниями. Активный уровень низкий. Указывает, что младший байт шины адреса содержит адрес устройства ввода/вывода. Кроме того, этот сигнал генерируется после выдачи подтверждения прерывания, тем самым указывая, что вектор прерывания может быть помещен на шину данных.

$\overline{\text{RD}}$ (чтение) – выход с тремя устойчивыми состояниями. Активный уровень низкий. Указывает, что микропроцессор готов к чтению данных из памяти или из устройства ввода/вывода. Адресованное устройство или память используют этот сигнал для стробирования при подаче данных на шины данных микропроцессора.

$\overline{\text{WR}}$ (запись) – выход с тремя устойчивыми состояниями. Активный уровень низкий. Указывает, что на шине данных содержатся данные, предназначенные для записи в память или вывода на устройство вывода.

$\overline{\text{RFSH}}$ (регенерация) – выход, активный уровень низкий. Указывает, что младшие семь разрядов шины адреса содержат адрес регенерации для ОЗУ и текущий сигнал $\overline{\text{MREQ}}$ должен использоваться для восстановления динамической памяти.

$\overline{\text{HALT}}$ (останов) – выход, активный уровень низкий. Указывает, что микропроцессор выполнил команду останова и ожидает появления немаскируемого либо маскируемого прерывания, после которого он продолжит работу. В состоянии останова микропроцессор будет выполнять холостые команды для обеспечения регенерации памяти.

$\overline{\text{WAIT}}$ (ожидание) – вход, активный уровень низкий. Используется медленными устройствами ввода/вывода и памятью для перевода микропроцессора в состояние ожидания до их готовности к передаче данных.

$\overline{\text{INT}}$ (запрос на маскируемое прерывание) – вход, активный уровень низкий. Может быть запрещено и разрешено состояние соответствующими командами микропроцессора. В ПЭВМ "Спектрум" каждые 20 мс на вход $\overline{\text{INT}}$ поступает аппаратно сформиро-

Таблица 2.1

Семейство микропроцессоров Z80

Тип	Разрядность слова данных/команд, бит	Технология	Емкость непосредственно адресуемого ЗУ, слов	Число основных команд	Максимальная тактовая частота МГц	Время выполнения команды наименьшее/наибольшее, мкс	Число уровней прерывания	Число регистров общего назначения	Встроенный тактовый генератор	Число выводов корпуса	Напряжение питания, В	Примечание
Z80	8	n-МОП	64 К	158	2,5	–	3	–	–	40	5	Программная совместимость с микропроцессором 8080А
Z80A	8	n-МОП	64 К	158	4	–	3	–	–	40	5	
Z80B	8	n-МОП	64 К	158	6	0,7/4,2	3	–	–	40	5	Модификация с пониженным потреблением мощности
Z80L	8	n-МОП	64 К	158	2,5	–	3	–	–	40	5	
Z80H	8/8	n-МОП	64 К	150	8	0,7/3,5	1	14	–	40	5	–
Z8108	8/16	n-МОП	512 К	–	25	0,4	7	18	+	40	5	Улучшенный вариант Z80 с устройством управления памятью.
NSC800	8	КМОП	64 К	158	2,5	1,6/9,2	5	–	+	40	5	Имеет систему команд микропроцессора Z80. Шинная структура аналогична 8085

ванный отрицательный импульс, по которому микропроцессор переходит к выполнению программы обслуживания клавиатуры, инкрементирует часы и т. д.

$\overline{\text{NMI}}$ (запрос на немаскируемое прерывание) – вход, активный уровень низкий Это прерывание имеет более высокий приоритет, чем сигнал $\overline{\text{INT}}$. Отрицательный фронт на этом входе активизирует внутренний триггер NMI и автоматически принуждает микропроцессор произвести повторный запуск (RESTART) с адреса 102 (0066 HEX).

$\overline{\text{RESET}}$ (сброс) – вход, активный уровень низкий. При поступлении этого сигнала обнуляется счетчик команд, регистры I и R, устанавливается режим прерывания "0" и сбрасывается триггер разрешения прерывания. Во время присутствия сигнала "RESET" шина адреса и шина данных находятся в третьем состоянии, а все сигналы управления – неактивны.

$\overline{\text{BUSRQ}}$ (запрос шин) – вход, активный уровень низкий. Этот сигнал имеет более высокий приоритет, чем сигнал $\overline{\text{NMI}}$ и всегда распознается в конце текущего машинного цикла. Он используется для организации прямого доступа к памяти (ПДП) и переводит в состояние высокого сопротивления все шины и тристабильные выходы сигналов управления, после чего шинами могут управлять другие внешние устройства.

$\overline{\text{BUSAK}}$ (подтверждение перевода шин в третье состояние) – выход, активный уровень низкий. Подтверждает передачу управления шинами запрашивающему внешнему устройству.

Ф – тактовый вход внешнего синхрогенератора.

Параметры микропроцессоров Z80 и совместимых с ним по системе команд приведены в табл. 2.1.

2.3. Программное обеспечение компьютера

2.3.1. Организация памяти компьютера

Для возможности работы операционной системы компьютера его память разбита на несколько областей. Границы этих областей могут быть фиксированными (постоянными) или изменяться в зависимости от конкретных требований.

Ниже приводится распределение памяти компьютера "Спектрум", где фиксированные границы обозначены числом, а переменные – именем системной переменной, в которой они хранятся [15, 22].

0 ... 16383 – ПЗУ.

16384 ... 22527	– экранная область. Изображение хранится здесь в черно-белом виде.
22528 ... 23295	– область атрибутов. Цвета, признаки яркости и мигания.
23296 ... 23551	– область буфера ZX-принтера. При отсутствии принтера часто используется для хранения небольших процедур в машинных кодах. Здесь они не повреждаются Бейсиком, а размещение их в нижней половине памяти повышает их быстродействие.
23552 ... 23733	– область системных переменных.
23734 ... CHANS-1	– карты памяти микродрайва. При работе с микродрайвом здесь хранится информация, например, об испорченных секторах на ленте и т.п. Если микродрайв отсутствует, то эта область не организуется.
CHANS ... PROG-2	– здесь размещается информация о каналах.
PROG...VARS-1	– программа на Бейсике.
VARS ... ELINE-2	– переменные Бейсик-программы.
ELINE ... WORKSP-1	–редактируемые строки программы.
WORKSP ... STKBON-1	–буфер инструкции INPUT.
STKBOT ... STKEND-1	– стек калькулятора. К нему обращается интерпретатор Бейсика. Калькулятор может вызываться из машинного кода командой ассемблера RST40. По окончании операции на вершину стека помещается ее результат.
STKEND ... sp	– свободное пространство. Сюда развиваются навстречу друг другу стек калькулятора и машинный стек. sp – это не системная переменная, а указатель машинного стека. Его можно вызвать только из машинного кода.
sp ... ERR SP	– машинный стек. Нужен процессору для выполнения обработки данных, адресов и пр.
ERR SP+1 ... RAMTOP	–стек возвратов. Здесь хранится информация о номере строки, к которой надо будет вернуться после выполнения подпрограммы.
RAMTOP ... UDG	– свободная область. Может быть использована для хранения процедур в машинных кодах.
UDG .. PRAMT	– область графики пользователя.
PRAMT	– физическая вершина памяти компьютера. Для "Спектрум -48" – PRAMT=65535, а для "Спектрум-16" – 32767.

2.3.2. Системные переменные

Компьютер выделяет место в оперативной памяти от адреса 23552 до адреса 23733 для размещения в нем некоторых переменных, которые нужны для нормального функционирования компьютера [15]. Однако пользователь может использовать информацию, содержащуюся в них для повышения эффективности программирования.

Первоначальные значения системных переменных устанавливаются после включения компьютера. Их устанавливают те процедуры, которые содержатся в ПЗУ и стартуют после включения.

Проверить содержание однобайтной системной переменной можно с помощью команды:

PRINT PEEK ADDR, где ADDR – это ее адрес. А для двухбайтной переменной эта команда будет выглядеть так:

PRINT PEEK ADDR + 256 * PEEK (ADDR + 1)

Можно заслать какие-либо значения (от 0 до 255) в адрес системной переменной при помощи оператора POKE, но в некоторых случаях такое изменение будет иметь лишь кратковременный характер и предыдущее значение будет само восстановлено (такие системные переменные обозначим буквой N), а в некоторых случаях изменение приведет к сбросу компьютера (такие переменные обозначим буквой X). Число в обозначении переменной указывает на длину переменной в байтах (для двухбайтовых переменных младший байт – первый). Характеристики переменных даны в табл. 2.2 [15, 22].

Таблица 2.2

Системные переменные

Тип, длина	Адрес	Имя	Содержание
1	2	3	4
N8	23552	KSTATE	Используется при сканировании клавиатуры.
N1	23560	LAST K	Запоминает последнюю нажатую клавишу.
1	23561	REPDEL	Время, в 50-х долях секунды, в течение которого надо удерживать нажатую клавишу, чтобы произошел автоповтор. Начальное значение – 35, но может быть изменено.
1	23562	PEPPER	Задержка, в 50-х долях секунды, между последовательными опросами клавиш. Начальное значение – 5.
N2	23563	DEFADD	Адрес аргументов функций пользователя если они используются. Иначе – Ø.

1	2	3	4
N1	23565	K DATA	Второй байт кода управления цветом при вводе с клавиатуры.
N2	23566	TV DATA	Здесь хранится информация о цвете и о параметрах AT и TAB для телевизора.
X38	23568	STRMS	Адреса каналов, подключенных к потокам.
2	23606	CHARS	Адрес, который лежит на 256 байтов ниже, чем набор символов, начинающийся с символа "пробел" и кончающийся символом " c ". Обычно этот набор находится в ПЗУ, но можно организовать свой набор в ОЗУ и изменить CHARS, чтобы она на него указывала.
1	23608	RASP	Продолжительность предупредительного звукового сигнала.
1	23609	PIP	Продолжительность звукового щелчка при нажатии клавиши.
1	23610	ERR NR	На единицу меньше, чем код сообщения об ошибке. Исходное значение – 255.
X1	23611	FLAGS	Различные флаги, необходимые для работы операционной системы.
X1	23612	TV FLAG	Флаги, связанные с работой телевизора.
X2	23613	ERR SP	Адрес в аппаратном стеке, используемый как адрес возврата при ошибке.
N2	23615	LIST SP	Адрес возврата после автоматического листинга.
N1	23617	MODE	Указывает на тип курсора "K", "L", "C", "E" или "G".
2	23618	NEW PPC	Номер строки, к которой следует переход
1	23620	NS PPC	Номер оператора в строке, к которому следует переход. Заслав нужное значение в NEW PPC, а затем в NS PPC, можно вызвать переход к оператору в нужной строке.
2	23621	PPC	Номер строки, в которой расположен исполняемый в данный момент оператор.
1	23623	SUB PPC	Номер оператора в строке, который в данный момент исполняется.
1	23624	BORDCR	Цвет бордюра, умноженный на 8. Здесь же содержатся атрибуты нижней части экрана.
2	23625	E PPC	Номер текущей строки (в которой находится курсор).
X2	23627	VARS	Адрес программных переменных.

1	2	3	4
N2	23629	DEST	Адрес переменной при исполнении оператора присвоения.
X2	23631	CHANS	Адрес канала данных.
X2	23633	CURCHL	Адрес данных для ввода–вывода.
X2	23635	PROG	Адрес БЕЙСИК-программы.
X2	23637	NXTLIN	Адрес следующей строки программы.
X2	23639	DATADD	Адрес разделителя после последнего введенного параметра из блока DATA.
X2	23641	E LINE	Адрес вводимой команды.
2	23643	K CUR	Адрес курсора.
X2	23645	CH ADD	Адрес следующего интерпретируемого символа.
2	23647	X PTR	Адрес символа после маркера "?".
X2	23649	WORK SP	Адрес рабочего пространства.
X2	23651	STK BOT	Адрес основания стека калькулятора.
X2	23653	STK END	Адрес вершины стека калькулятора.
N1	23655	BREG	Содержимое регистра "B" калькулятора.
N2	23656	MEM	Адрес области, используемой как память калькулятора (обычно совпадает с MEMBOT, но не всегда).
1	23658	FLAGS2	Флаги.
X1	23659	DF SZ	Число строк, включая и одну чистую в нижне части экрана (в системном окне).
2	23660	S TOP	Номер верхней программной строки при автоматическом листинге.
2	23662	OLDPPC	Номер строки, на которую указывает CONTIN
1	23664	OSPPC	Номер оператора в строке, к которому выполняется переход по CONTINUE.
N1	23665	FLAGX	Вспомогательные флаги.
N2	23666	STR LEN	Длина строковой переменной (стринга) в операторе присвоения.
N2	23668	T ADDR	Адрес следующего параметра в синтаксической таблице.
2	23670	SEED	Начальное значение для RND, изменяется функцией RANDOMIZE N.
3	23672	FRAMES	Трехбайтная переменная (старший байт первый). Получает приращение на единицу каждую пятидесятую долю секунды. Может

Продолжение табл. 2.2

			использоваться для организации контроля времени. Так, команда .PRINT (256*256*PEEK 23674 + 256*PEEK 23673 + + PEEK 23672)/50 покажет, сколько секунд прошло со времени включения компьютера.
2	23675	UDG	Адрес 1-го символа графики пользователя.
1	23677	COORDS	Координата X последней помещенной на экран точки.
1	23678		Координата Y последней помещенной на экран точки.
1	23679	P POSN	Номер колонки позиции, в которой находится принтер (всего 33 позиции).
1	23680	PR CC	Младший байт адреса следующей позиции печати по LPRINT (в буфере принтера).
1	23681		Не используется.
2	23682	ECHO E	Номер колонки и номер строки конца буфера клавиатуры (всего 33 позиции, 24 строки).
2	23684	DF CC	Адрес в экранной области позиции печати.
2	23686	DF CCL	Подобно DF CC для нижней части экрана.
X1	23688	S POSN	Номер колонки позиции печати (33).
X1	23689		Номер строки позиции печати (24).
X2	23690	S POSN	Подобно S POCNL для нижней части экрана
1	23692	SCR CT	На единицу больше, чем необходимое количество передвижек строк на экране до остановки с запросом SCROLL? Если организовать постоянную подачу после PRINT какого-либо числа, большего, чем 1, в эту ячейку, то печать будет идти непрерывно, без запроса SCROLL?
1	23693	ATTR P	Текущие цветовые атрибуты (для всего экрана).
1	23694	MASK P	Маска локальных атрибутов. Каждый бит этой переменной, если он равен единице, показывает, что соответствующий цветовой атрибут надо брать не из ATTR P, а из того, что уже есть на экране в позиции, для которой выполняется печать.
N1	23695	ATTR T	Временные атрибуты. Их установка справедлива только для выполнения текущего оператора PRINT, INPUT.

1	2	3
SIN	LET A = SIN 2.345	Синус угла заданного в радиана
SQR	PRINT SQR 25	Квадратный корень.
STR$	PRINT STR$ 100	Перевод числа в строковую пер менную (стринг), изображающу то же число. STR$ 100 = "100".
TAN	LET A = TAN 2.345	Тангенс угла, заданного в рад анах.
USR аддр	RANDOMIZE USR 30000	Вызывает программу в машинно коде и стартует ее с указанног адреса.
USR A$	LET аддр=USR "A"	Дает адрес в области график пользователя UDG, в котором ра положено изображение символ графики пользователя, соотве ствующего "A".
VAL	LET A=VAL "100"	Функция, обратная STR$, обеспечивает перевод стринга в соотве ствующее число.
VAL$	PRINT VAL $ """"100""""	Рассматривает свой стринговы параметр без ограничительных к вычек как новый стринг.
	Операторы	
BEEP	BEEP 20.0, 0.5	Воспроизводит в динамике ко пьютера звуковой сигнал (пр должительность – 20 с, высота на 0,5 тона выше, чем звук "д первой октавы).
BORDER	BORDER N	Установка цвета бордюра. N = 0, ..., N.
BRIGHT	BRIGHT i	Установка яркости: 0 – нормальн 1 – повышенная.
CAT		Распечатать каталог картридж микродрайва.
CIRCLE	CIRCLE X, Y, R	Вычерчивание окружности с цен ром X,Y и с радиусом R.
CLEAR	CLEAR аддр	Вычищает все переменные, выпо няет RESTORE и CLS, вычищает стек GO SUB. Устанавливает

1	2	3
		вую верхнюю границу памяти по адресу аддр.
CLOSE#		Команда интерфейса-1 (закрыть поток).
CONTINUE		Продолжить исполнение программы после прерывания.
COPY		Команда ZX-принтера. Выполняет копирование экрана на бумагу.
DATA		Данные для оператора READ.
DEF FN	DEF FN W(X, Y,...)	Задает функцию пользователя. Имя функции W – однобуквенное, а параметры функции должны стоять в скобках (если параметры отсутствуют, то скобки все равно должны быть).
DIM	DIM A (i, j, k)	Задает 1÷3-мерный массив А, элементами которого являются индексируемые переменные Ai, j, k.
	DIM A$ (i, j, k)	Задает символьный 1÷3-мерный массив А, элементами которого являются строки символов фиксированной длины. Исходное значение элементов массива устанавливается в " " (пробел).
DRAW	DRAW X,Y	Вычерчивается прямая линия от точки, в которой производилась печать, до точки, отстоящей от нее по горизонтали на X и по вертикали на Y.
	DRAW X,Y,Z	Вычерчивается дуга окружности, охватывающая угол Z (измеряемый в радианах), соединяющая точку, в которой производилась печать, до точки, отстоящей от нее по горизонтали на X и по вертикали на Y.
ERASE		Команда микродрайва.
FLASH	FLASH 1	Признак мигания: 1 – включено, 0 – выключено.

1	2	3
FOR	FOR i=x TO y	Составная часть оператора FOR ... TO ... NEXT.
	FOR i=x TO y STEPz	Организует цикл с соответству щими начальным и конечным зн чениями параметров x, y и шаг z. Шаг может быть положительны или отрицательным (в последне случае x > y).
FORMAT		Команда микродрайва.
GO SUB	GO SUB N	Переход к исполнению подпр граммы, начинающейся со стро N (подпрограмма завершается к мандой RETURN).
GO TO	GO TO N	Переход к исполнению строки N.
IF X THEN Y	IF A>5 THEN GO TO N	Условный оператор. Если выраж ние X справедливо, то выполн ется выражение после THEN. В пр тивном случае выполняется пер ход к следующей строке (а не следующему оператору).
INK	INK N	Устанавливает цвет печат N = 0, ..., 7.
INPUT	INPUT A	Обеспечивает ввод с клавиатур числа во время исполнения пр граммы. При вводе символьно стринга этот оператор имеет в INPUT LINE.
INVERSE	INVERSE Ø	Признак инверсии: 0 – нормальн печать, 1 – печать белым черному.
LET	LET A = X	Оператор присваивания; перем ной A присваивается значение
LIST	LIST	Отображается текст (листинг) пр граммы на экране дисплея (тел визора). В формате LIST N текст программы выдается, начиная строки N. Если текст програм занимает более одного экрана, после заполнения экрана выдает запрос SCROLL? В этом случ

1	2	3
		нажатие клавиши "BREAK" или "N" прерывает распечатку текста, а нажатие любой другой клавиши клавиатуры продолжит листинг программы.
LLIST	LLIST N	Команда ZX-принтера. Выполняется аналогично предыдущему оператору LIST, за исключением того, что текст программы печатается на принтере ПЭВМ.
LOAD	LOAD "имя"	Загрузка БЕЙСИК-программы с магнитофона. Если имя не указано LOAD""), то загрузится первая встречная программа (Внимание: между кавычками не должно быть пробела).
	LOAD "имя" CODE M,N	Команда на загрузку блока программы в машинных кодах (загружает N байтов, начиная с адреса M). В случае отсутствия параметров M, N блок программы будет загружен в те адреса из которых он выгружен.
	LOAD"имя"DATA( )	Загрузка блока данных.
	LOAD"имя"DATA$( )	Загрузка символьного массива.
	LOAD"имя"SCREEN$	Загрузка экрана. Функция аналогично оператору LOAD "имя" CODE 16384, 6912. Экранная область памяти ПЭВМ "Спектрум" всегда начинается с адреса 16384 и имеет 6912 байтов (6144 – графика, 768 – атрибуты).
LPRINT		Команда ZX-принтера (см. оператор PRINT).
MERGE	MERGE "имя"	Аналогично LOAD, но загружаемая программа не уничтожает старую.
MOVE		Команда микродрайва.
NEXT		Заканчивает цикл. Часть оператора FOR... TO...NEXT.

1	2	3
OPEN#		Команда интерфейса-1 и микродрайва.
OUT	OUT M,N	Выдает байт N информации на внеш ний порт M.
OVER	OVER 1	Управляет условием наложения символов друг на друга при печати: 0 – новый символ полностью закрывает старый, 1 – изображение создается смешением символов.
PAPER	PAPER N(N=0...7)	Задает цвет фона (аналогично INK).
PAUSE	PAUSE N	Пауза в работе программы на время N/50 с или до нажатия какой-либо клавиши.
	PAUSE Ø	Пауза до нажатия клавиши.
PLOT	PLOT C,M,N	Выполняет печать точки на экране дисплея (телевизора) с цветом C и координатами M, N (M=0,..., 255, N=0,..., 175). Если параметр C не задан, то устанавливается текущий цвет. Обычно эта команда предшествует команде DRAW.
POKE	POKE аддр, N	Засылает байт по указанному адресу.
PRINT	PRINT N	Печать числа.
	PRINT X	Печать результата расчета выражения X.
	PRINT "ABCD"	Печать текста, стоящего в кавычках.
	PRINT A$	Печать стринга (строковой переменной). Одной командой PRINT может выполняться печать сразу нескольких переменных или чисел, разделяемых любым из трех разделителей: запятой (,), точкой с запятой (;), апострофом ('). При использовании запятой печать выполняется на той же строке, начиная 16-й колонки (в правой половин

1	2	3
		экрана). При использовании точки с запятой печать выполняется в той же строке без зазора, а при использовании апострофа происходит переход к новой строке печати. Пример записи: PRINT A,B,C; D; E$'F$'N. Просто PRINT дает пустую строку. Например, PRINT, PRINT, PRINT эквивалентно PRINT.
	PRINT AT M,N	Печать в заданной позиции (строка, столбец.
	PRINT TAB N	Печать с табуляцией (N). N – величина табуляции. Оператор PRINT может сопровождаться операторами задания цветовых атрибутов INK, PAPER, FLASH, BRIGHT, INVERSE, OVER (вводятся через точку с запятой). Пример записи: PRINT INK 6; PAPER 1; "VERY WELL", FLASH 1.
RANDOMIZE N		Устанавливает системную переменную SEED, используемую для работы генератора псевдослучайных чисел (см. RND).
RANDOMIZE		Соответствует RANDOMIZE Ø; (N= = 0, ..., 65535).
RANDOMIZE USR	RANDOMIZE USR аддр	Вызывает и запускает процедуру, записанную в машинных кодах, находящуюся по указанному адресу. RANDOMIZE USR Ø инициализирует компьютер, т.е. выполняет эффективный сброс, эквивалентный выключению питания.
READ	READ V1,V2, V3...VN	Присваивает переменным значения, находящиеся в соответствующем операторе DATA
REM		Пустой оператор, открывающий строку комментариев. Все записанное за оператором REM (в пределах строки) не исполняется, а

1	2	3
		только распечатывается (пр исполнении оператора LIST).
RESTORE N		Устанавливает указатель DATA, привязывающий оператор DATA к соответствующему оператору READ, указывающему на операто DATA, расположенному в строке N (или к ближайшей старшей).
RESTORE		Аналогично, что и RESTORE Ø.
RETURN		Пустой оператор, завершающий подпрограмму, к которой обращались по GO SUB. Передает управление в строке, следующей за вызываемой.
RUN	RUN N	Команда на исполнение программы, по которой все переменные существовавшие до этой программы, уничтожаются и начинаетс исполнение со строки N. Эквивалентна CLEAR : GO TO N.
	RUN	Эквивалентна команде RUN Ø
SAVE	SAVE "имя"	Выполняется запись БЕЙСИК-программы на магнитную ленту магнитофона. Требует задания имени и нажатия клавиши "ENTER", после чего на экране дисплея появляется "START TAPE THEN PRESS ANY KEY" (запустите ленту и нажмите любую клавишу).
	SAVE "имя" LINE N	Записывает программу на ленту так, что она после загрузки будет автостартовать со строки N.
	SAVE "имя" CODE M,N	Обеспечивает запись на ленту блока машинных кодов, начиная с адреса M, в количестве N байтов
	SAVE "имя" DATA ( )	Обеспечивает запись числового массива.
	SAVE "имя" DATA$ ( )	Обеспечивает запись на ленту символьного массива.
	SAVE "имя" SCREEN$	Обеспечивает запись экрана. Экви-

Окончание табл. 2.3

1	2	3
		валентна команде SAVE "имя" CODE 16384, 6912. (Внимание: наличие всех параметров в команде SAVE обязательно).
STOP		Останавливает исполнение программы. Продолжить работу программы можно оператором CONTINUE.
VERIFY	VERIFY " "	Аналогична команде LOAD (имеет все те же формы записи), но в отличие от нее не производит загрузку информации в ОЗУ, а только сравнивает то, что записано на магнитной ленте с тем, что находится в памяти компьютера и в случае несоответствия выдает сообщение об ошибке. Этот оператор применяется для проверки качества записи.

Клавиатура. Режимы работы. Курсоры.

Клавиатура ПЭВМ "Спектрум" содержит всего 40 клавиш, но каждая из них имеет многофункциональное значение и позволяет выполнять по 6-7 различных действий, вводить как отдельные символы, так и целые слова. Действие, производимое клавишей определяется частично переключающими (префиксными) клавишами "CAPS SHIFT" и "SYMBOL SHIFT", а частично режимом, в котором находится компьютер. Режим отображается курсором – мерцающей буквой, указывающей позицию, в которую вводится очередной символ с клавиатуры. Внешний вид клавиатуры дан на рис. 2.4 [4].

Рассмотрим функции некоторых характерных клавиш [15, 22]:

"ENTER" – нажатие этой клавиши обычно завершает ввод в компьютер чего-либо (команды, данных, программной строки). На компьютерах других систем эта клавиша аналогична клавишам "RETURN", ВК (возврат каретки) и т. п.

"CAPS SHIFT" – переключение регистра на печать прописными буквами.

"SYMBOL SHIFT" – переключение регистра для печати символов (+, –, , и т.д.), а также некоторых ключевых слов (THEN, TO, OR, AND, STEP и др.).

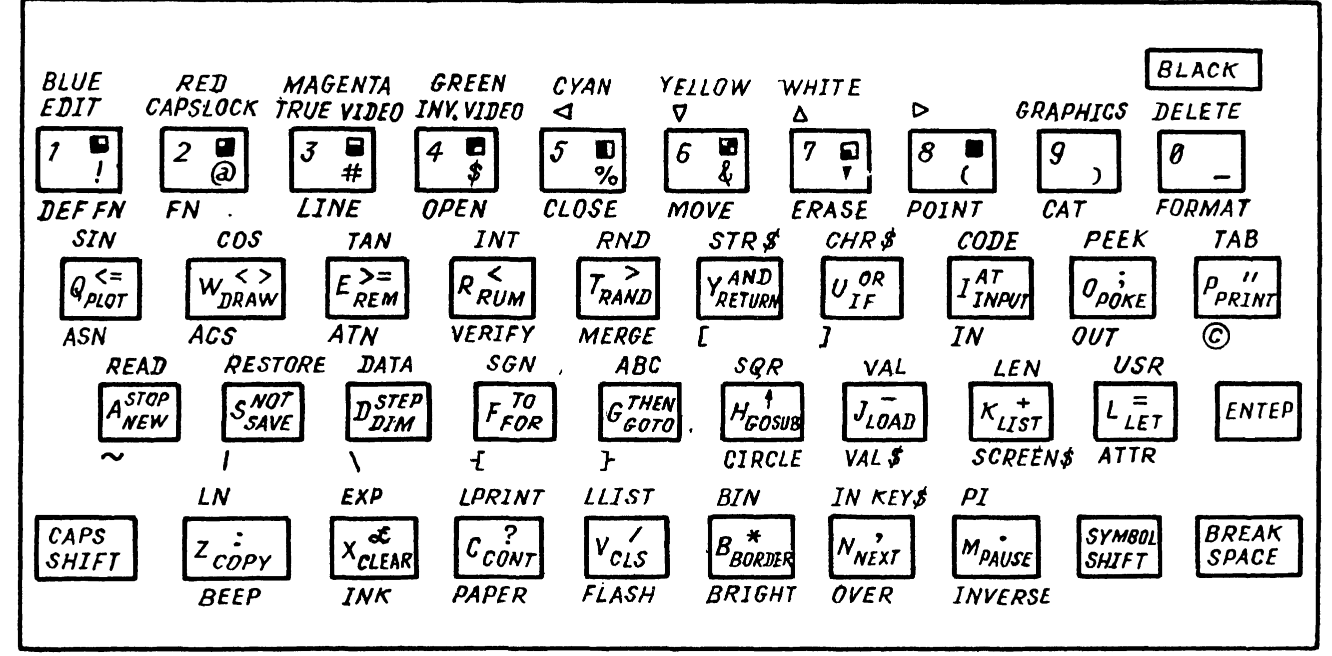


Рис. 2.4. Внешний вид клавиатуры ПЭВМ "Спектрум"

"SPACE" – пробел. Эта же клавиша выполняет команду BREAK (прерывание исполнения программы), если ее нажать вместе с клавишей "CAPS SHIFT".

Остальные клавиши имеют буквенное или цифровое обозначение.

При включении компьютера на экране появляется исходное сообщение: "1982 Sinclair Research Ltd". Это означает, что первичные проверки в компьютере прошли нормально и, скорее всего, он исправен и готов к работе.

При нажатии клавиши "ENTER" на экране появится черный квадрат с мерцающим буквенным обозначением на нем. Это курсор. Во-первых, он показывает то место на экране, в котором будет помещаться очередной набранный символ, а во-вторых, указывает, в каком режиме находится клавиатура. Этих режимов четыре, и им соответствуют пять разных курсоров.

Командный режим. Курсор "K". Он означает, что сейчас при нажатии клавиши будет введена команда, которая закреплена за этой клавишей, или цифра. Компьютер сам "предполагает", что строка может начинаться только либо с номера строки, либо с команды, поэтому выдается курсор "K". После того как команда набрана и дальше должны пойти ее параметры, режим автоматически переключается на литерный (курсор "L"). Если вводится ":" (двоеточие), то курсор опять переключается на "K", т.к. двоеточие является разделителем между несколькими командами, если они записываются в одной строке.

Литерный режим. Курсоры "L" и "C". Курсор "L" соответствует печати строчными буквами, а курсор "C" – прописными. Переключиться на курсор "C" можно командой CAPS LOCK. Это выполняется одновременным нажатием "CAPS SHIFT" и цифры "2". В этом же режиме набираются символы и служебные слова, связанные с клавишей "SYMBOL SHIFT".

Расширенный командный режим. Курсор "E". В этом режиме набираются команды, записанные под клавишами или над клавишами. Переход в режим "E" выполняется одновременно нажатием клавиш "CAPS SHIFT" и "SIMBOL SHIFT". Он действует только на одно нажатие. Если в режиме "E" нажать какую-либо клавишу, то появится слово, записанное над клавишей а если нажать эту клавишу совместно с "CAPS SHIFT", то появится слово, записанное под клавишей. Исключение составляют цифровые клавиши (верхний ряд). Чтобы набрать слово, записанное под клавишей, надо в режиме "E" нажать не "CAPS SHIFT", а "SYMBOL SHIFT" совместно с клавишей.

Графический режим. Курсор "G'. В этом режиме набираются символы блочной графики, расположенные на цифровых клавишах, а также символы графики пользователя (UDG – user defined graphics).

Переход в графический режим выполняется одновременным нажатием клавиш "CAPS SHIFT" и цифры "9". Если в этом режиме нажимается, например, клавиша "5", то на экране появится квадрат, левая половина которого черная, а правая–белая. Нажатие клавиши "5" совместно с клавишей "CAPS SHIFT" изобразит тот же квадрат в инвертированном виде, т.е. левая половина будет белая, а правая – черная. Выход из графического режима выполняется так же, как и вход.

Таким образом, клавиатура компьютера обеспечивает ввод большинства команд расширенной версии Бейсика нажатием одной-двух клавиш. Эта уникальная способность ПЭВМ "Спектрум" вводить операторы и функции не по буквам, а одним нажатием клавиши, называется токенизированной формой записи ключевых слов. Сначала это выглядит несколько сложновато, но опыт приходит быстро. Если набрана произвольная или ошибочная строка, то при нажатии клавиши "ENTER" на экране появится "?". Чтобы начать работу, надо эту строку стереть. Это выполняется командой DELETE (одновременное нажатие "CAPS SHIFT" и цифры "0"). Когда все символы будут уничтожены, курсор встанет на исходную позицию и примет вид "K", т.е. компьютер ожидает команду.

При нажатии клавиши "R" на экране появится команда RUN. Курсор изменится на "L". При повторном нажатии той же клавиши появится буква r. Если клавишу задержать в нажатом положении, то буква будет повторена многократно. Это удобная функция называется "автоповтор" (кстати, он действует и при стирании).

Нажатие этой клавиши совместно с "CAPS SHIFT" даст прописную букву R, а совместно с "SYMBOL SHIFT" – знак "<" .

Переход в расширенный командный режим – использование клавиш "CAPS SHIFT" и "SYMBOL SHIFT" одновременно. В этом случае появится курсор "E". Теперь нажатие той же клавиши даст оператор INT. Необходимо войти в режим "E" и нажать клавишу совместно с "CAPS SHIFT". Получится команда VERIFY.

При работе с компьютерами "Спектрум+", "Спектрум+128", "Спектрум+2" и "Спектрум+3" можно упростить работу благодаря наличию дополнительных клавиш, которые выполняют ряд функций одним нажатием:

"DELETE"	–стирание символа;
"GRAPH"	– переход в графический режим;
"EXTENDED MODE"	– переход в расширенный командный режим;
"EDIT"	– редактирование строки,
"CAPS LOCK"	– переключение регистра на печать прописными буквами;
"INV. VIDEO"	– включение инверсного режима (печать белым по черному);
"TRUE VIDEO"	– возвращение из инверсного режима.

Кроме того там имеются дополнительные клавиши для набора знаков препинания: точка, запятая, точка с запятой, кавычки, а также четыре клавиши для управления перемещением курсора при редактировании.

В табл. 2.4 приводится расписание клавиатуры ПЭВМ "Спектрум" [15], которое, несомненно, будет полезно тем, кто работает с самодельными компьютерами и лишен возможности иметь замечательную инструкцию, прилагающуюся к ПЭВМ "Спектрум-48".

Таблица 2.4

Расписание клавиатуры компьютера [15]

Клавиша	Курсор					
	K	L			E	
			CAPS SHIFT	SYMBOL SHIFT		CAPS SHIFT
1	2	3	4	5	6	7
A	NEW	a	A	STOP	READ	~
B	BORDER	b	B	*	BIN	BRIGHT
C	CONTINUE	c	C	?	LPRINT	PAPER
D	DIM	d	D	STEP	DATA	\
E	REM	e	E	>=	TAN	ATN
F	FOR	f	F	TO	SGN	{
G	GO TO	g	G	THEN	ABS	}
H	GO SUB	h	H	↑	SQR	CIRCLE
I	INPUT	i	I	AT	CODE	IN
J	LOAD	j	J	–	VAL	VAL $
K	LIST	k	K	+	LEN	SCREEN$
L	LET	l	L	=	USR	ATTR
M	PAUSE	m	M		PI	INVERSE
N	NEXT	n	N	,	INKEY$	OVER
O	POKE	o	O	;	PEEK	OUT
P	PRINT	p	P	”	TAB	©
Q	PLOT	q	Q	<=	SIN	ASN
R	RUN	r	R	<	INT	VERIFY
S	SAVE	s	S	NOT	RESTORE	|
T	RANDOMIZE	t	T	>	RND	MERGE
U	IF	u	U	OR	CHR$	]
V	CLS	v	V	/	LLIST	FLASH
W	DRAW	w	W	<>	COS	ACS

1	2	3	4	5	6	7	
X	CLEAR	x	X	£	EXP	INK	
Y	RETURN	y	Y	AND	STR$	[	
Z	COPY	z	Z	:	LN	BEEP	
0	0	0	DELETE	_		FORMAT	*
1	1	1	EDIT	!		DEF FN	*
2	2	2	CAPS LOCK	@		FN	*
3	3	3	TRUE VIDEO	#		LINE	*
4	4	4	INV. VIDEO	$		OPEN#	*
5	5	5	←	%		CLOSE#	*
6	6	6	↓	&		MOVE	*
7	7	7	↑	'		ERASE	*
8	8	8	→	(		POINT	*
9	9	9	GRAPHICS	)		CAT	*

* Символы верхнего ряда (цифровые клавиши) в расширенном командном режиме (курсор "E") нажимаются не с "CAPS SHIFT", с "SYMBOL SHIFT".

Составление и редактирование программ

ПЭВМ "Спектрум" может работать в режиме непосредственного исполнения команд и программном режиме [15].

В первом режиме команды выполняются после нажатия клавиши "ENTER" и после исполнения в памяти компьютера не хранятся. В случае необходимости повторения команды она снова набирается с клавиатуры (PRINT 2*2 <ENTER>4 O.K. – операция вычисления 2×2 = 4, комментарий O.K. – о'кей). Прямая команда может состоять из нескольких команд, которые разделяются двоеточием (например, создание таблицы квадратов целых чисел от 1 до 20 FOR i = 1 TO 20: PRINT i, i * i: NEXT i) [15].

Во втором режиме команды записываются в виде строк программы, начинающихся с номера. Строки в программе размещаются в порядке возрастания номеров. Как правило, строки программы нумеруют числами через десяток (10, 20, 30, ...). Это дает возможность легко вставить пропущенную строку в программу. Действительно, чтобы вставить строку между строками 1 и 2, их нужно

перенумеровать, а чтобы вставить строку между строками 10 и 20, достаточно новой строке присвоить номер от 11 до 19, например 15. В программе ПЭВМ "Спектрум" номер строки может быть в интервале от 1 до 9999. В одной строке могут быть записаны несколько команд или операторов, которые должны разделяться двоеточием

Если набрана строка с номера N, и строка с таким номером в программе уже имеется, то после нажатия клавиши "ENTER" новая строка встанет на свое место в программе, а старая будет уничтожена. Таким образом, для стирания ненужных строк достаточно набрать номер строки, подлежащей уничтожению, и нажать "ENTER".

Программа запускается командой RUN. После отработки программа остается в памяти (в отличие от прямой команды) и может быть многократно повторена. Исполнение работающей программы можно прервать командой BREAK. Если необходимо внести изменение в программу, то ее следует остановить (командой BREAK), затем командой LIST (или LIST N) вывести текст программы на экран и выполнить редактирование.

Так как ПЭВМ "Спектрум" имеет строчный программный редактор, пользователь может редактировать (переделывать) каждую строку отдельно. Для редактирования строки она вызывается в позицию редактирования, расположенную в нижней части экрана. Например, нужно устранить ошибку в работающей программе. После остановки программы (командой BREAK) дается команда LIST для печати текста на экране. Если текст программы длинный, то он весь не помещается на экране. Поэтому после заполнения текстом экрана распечатка прекратится, и появится запрос SCROLL ?. При наличии на экране строки, которую нужно переделать, следует в ответ на запрос нажать клавиши N (нет) или BREAK, а при отсутствии строки на экране – нажать любую другую клавишу, и распечатка текста программы на экране будет продолжаться до получения очередного запроса SCROLL ?.

Когда нужная строка найдена, необходимо подвести к ней курсор, имеющий вид стрелки ">" и перемещаемый курсорными клавишами ("CAPS SHIFT 6" – вниз; "CAPS SHIFT 7" – вверх). Если программа длинная, то целесообразнее сразу обратиться по номеру строки N командой LIST N и в ответ на запрос SCROLL ? нажать клавишу "BREAK". После этих операций курсор ">" сразу будет установлен против нужной строки.

Затем найденная строка вызывается в позицию редактирования при помощи ключевого слова EDIT (CAPS SHIFT 1). Строка повторяется в нижней части экрана. Здесь можно вносить в строку изменения. Для удаления из строки ненужных символов следует установить курсор в строке с помощью курсорных клавиш ("CAPS SHIFT 5" – влево; "CAPS SHIFT 8" – вправо) справа от символа и

нажать клавишу "DELETE" ("CAPS SHIFT 0 "). Ввод новых символов и ключевых слов производится по месту установки курсора. Если отредактированная строка не содержит синтаксических ошибок, то после нажатия клавиши "ENTER" она займет свое место в программе. При необходимости вызова фрагмента программы, который последним редактировался или вводился, нужно сразу нажать клавишу "ENTER". Такой прием сокращает время поиска нужной строки.

Существуют другие модификации редактора версии Бейсика, например Бета Бейсик (версии 1.0; 1.8, 3.0), загружаемой с кассеты. Эта версия обеспечивает эффективное редактирование программ (имеет автонумерацию строк, возможность перенумерации строк, команду EDIT N для немедленного редактирования строки N, возможность быстрой переброски курсора, отладочные операторы, дополнительные операторы организации цикла, возможность работы с "окнами", развитые средства структурного программирования (процедуры и другие удобные функции и программы) [15].

Основы программирования [22]

Условия. Простейшим условием при выполнении программы может быть сравнение двух чисел или двух строк. При этом числа могут быть равны или одно больше другого, а строки равны или одна следует после другой в алфавитном порядке.

Знаки условия набираются на клавиатуре следующим образом:

> (больше) – "SYMBOL SHIFT" вместе с "T";

< (меньше) – "SYMBOL SHIFT" вместе с "R";

<= (меньше или равно) – "SYMBOL SHIFT" вместе с "Q" (нельзя набирать < и =);

>= (больше или равно) – "SYMBOL SHIFT" вместе с "E";

<> (не равно) – "SYMBOL SHIFT" вместе с "W"

Например, выражения 2 < 3, 5 > –8 ИСТИННЫ, а выражения 7 > 9, 2 < 1 ЛОЖНЫ.

В определенных местах программы компьютер должен принимать решение о дальнейшем ходе вычислений. При выполнении условий используется оператор, реализующий форму:

IF – некоторое истинное или ложное выражение, THEN – некоторое действие.

Ниже приводится программа игры для двух человек, реализующая оператор условия IF... THEN... (для реализации программы необходимо дать команду NEW, а затем набрать и выполнить программу):

```
10 REM GUESS THE NUMBEK (УГАДЫВАНИЕ ЧИСЛА)
20 INPUT A:CLS
30 INPUT "GUESS THE NUMBER ,B (УГАДАЙТЕ ЧИСЛО)
40 IF A=B THEN PRINT "THIS IS CORRECT" :STOP
50 IF B<A THEN PRINT "THIS IS TOO SMALL, TRY AGAIN"
60 IF B>A THEN PRINT "THIS IS TOO BIG, TRY AGAIN"
70 GOTO 30
```

Таким образом, оператор IF имеет форму

IF УСЛОВИЕ THEN...

где "..." – последовательность операторов, разделенных двоеточием. Если "УСЛОВИЕ" истинно, то выполняются операторы, следующие после THEN, в противном случае они игнорируются (пропускаются) и программа выполняется до следующего оператора.

Циклы. Значительно упрощают программу применения циклов при многократных вычислительных операциях. Для организации циклов используют оператор FOR, общая форма записи которого следующая:

FOR "УПРАВЛЯЕМАЯ ПЕРЕМЕННАЯ" = "НАЧАЛЬНЫЕ ЗНАЧЕНИЯ" TO "ПРЕДЕЛ" STEP "ШАГ ПРИРАЩЕНИЯ"

Здесь "НАЧАЛЬНЫЕ ЗНАЧЕНИЯ", "ПРЕДЕЛ", "ШАГ ПРИРАЩЕНИЯ" есть выражения, принимающие числовые значения.

Например, программа, подсчитывающая сумму вводимых пяти чисел, будет

```
10 LET TOTAL=0
20 FOR C=1 TO 5
30 INPUT A
40 REM C=NUMBER OF TIMES THAT A HAS BEEN INPUT SO FAR
50 LET TOTAL=TOTAL+A
60 NEXT C
 0 PRINT TOTAL
```

В этой программе "C" – управляющая переменная цикла (должна иметь имя в одну букву) "C" последовательно принимает значения 1, 2, 3, 4, 5 (предел–конечное значение управляющей переменной цикла) и при каждом проходе выполняются строки 30, 40, 50. После того как переменная "C" примет пятое значение, выполнится 70-я строка. В программе приращение значения переменной "C" составляет 1. Это значение можно изменить, используя указание STEP как часть оператора FOR.

Это демонстрирует программа, выводящая числа от 1 до 10 в убывающей последовательности.

```
10 FOR N=10 TO 1 STEP -1
20 PRINT N
30 NEXT N
```

Подпрограммы. Для удобства выполнения программы иногда целесообразно фрагменты программы представить в виде отдельных частей (подпрограмм), по несколько раз используемых в различных местах программы. Для вызова подпрограмм используются операторы GO SUB (GO TO SUBROUTINE) и RETURN в форме GO SUB N, где N – номер первой строки в подпрограмме.

При использовании этого оператора компьютер запоминает следующий после GO SUB оператор, которому и передается управление после выполнения подпрограммы. Реализуется это путем помещения номера оператора (адреса возврата) в специальную

область памяти, называемую GO SUB стек. Команда RETURN выбирает верхний адрес возврата из GO SUB стек и продолжает выполнение программы с оператора, следующего после оператора с этим номером Подпрограмма может вызывать другие подпрограммы и также саму себя (называется рекурсивной подпрограммой).

Операторы READ, DATA, RESTORE. Для экономии времени иногда можно использовать эти операторы вместо оператора INPUT.

Оператор READ состоит из слова "READ" и списка переменных, разделенных запятыми, особенно эффективно его использовать, когда вводимые значения присваиваются переменным. Например:

```
10 READ A,B,C
20 PRINT A,B,C
30 DATA 10,20,30
40 STOP
```

В этой программе компьютер ищет величины переменных в утверждении оператора DATA.

Оператор DATA не является управляющим и выполняется только в программе. Оператор DATA – это список значений (числовых или строчных), разделенных запятыми В процессе работы компьютер выбирает первое значение из списка оператора DATA для величины из оператора READ, затем он берет второе значение из списка оператора DATA. Таким образом, выбираемые последовательно операторы READ. обрабатываются из списка оператора DATA.

Оператор DATA может также содержать и строчные переменные.

Оператор RESTORE используется для организации программного перехода для DATA-списков В этом случае используется оператор RESTORE с указанием после него номера строки с оператором DATA и все последовательно встречающиеся в программе операторы READ вводят данные подряд, начиная с первого оператора DATA.

Арифметические операции. Кратко рассмотрим такие арифметические операции, как сложение (+), вычитание (–), умножение (*) и деление (/). Следует помнить, что операции умножения и деления имеют первый приоритет, а сложения и вычитания – второй. Относительно друг друга умножение и деление имеют равные приоритеты. Эти операции выполняются последовательно слева направо (а затем также выполняются операции сложения и вычитания). Для задания приоритета на компьютере используются числа от 1 до 16 Например, операции "*" "/" имеют приоритет 8, а "+", "–" – 6. Это жесткий порядок вычисления, его можно изменить с помощью скобок. В этом случае выражение в скобках вычисляется первым, а затем полученное число подставляется в общее выражение

Приведем пример арифметической операции:

LET TAX = SUM * 15/100 – 10

Это означает: значение переменной с именем "SUM" следует умножить на 15, разделить его на 100, а затем от полученного числа отнять 10.

Можно использовать операцию сложения для сцепления строк (конкатенции) в выражениях.

При работе с ПЭВМ следует помнить, что компьютер использует арифметику с плавающей точкой (запятой), при этом различные части числа (мантисса и порядок) хранятся в отдельных байтах. Это приводит иногда к неточным результатам даже для целых чисел.

Функции. Функции, используемые в Бейсике LEN, STR$, VAL, SGN, INT, SQR, DEF, FN – это "зашитые" в Бейсик-систему подпрограммы, которые, получая значения (аргументы), выдают результаты. Функции используются в математическом выражении простым включением в него имени функции с последующими аргументами. Кратко рассмотрим эти функции.

Функция LEN определяет длину заданного в ней строкового аргумента в символах.

Функция STR$ преобразует число в символьный код, аналогичный формату вывода числа оператором PRINT.

Функцию VAL можно использовать в двух случаях: во-первых, как функцию, позволяющую обратиться к другой функции (STR$) и преобразовать строку в число, во-вторых, как функцию VAL$, в которой и аргументом, и результатом функции является строка символов (она работает как VAL, применяемая дважды, раскрывая все кавычки в строках).

Функция SGN представляет собой так называемую математическую функцию сигнум (знак). В этой функции аргумент и результат–числа. Результат функции равен:

1, если аргумент положителен;

0 , если аргумент равен 0 ;

–1, если аргумент отрицателен.

Функция ABS преобразует аргумент в положительное число.

Функция INT преобразует дробное число в целое путем отбрасывания дробной части числа.

Функция SQR вычисляет корень квадратный из числа. Пользователь может сам определить для себя какую-либо функцию, указав FN и имя этой функции (буква при числовом аргументе; буква и $ при строковом аргументе) Аргументы в функции FN должны быть обязательно заключены в скобки. Можно также определить функцию вводом оператора DEF в некотором месте программы.

Математические функции. В языке Бейсик использованы следующие математические функции: возведение в степень **, вычисление значения π – PI, экспоненты EXP, логарифма LN, тригонометрических функций SIN, COS, TAN, ASN, ACS, ATN. Запись таких выражений на Бейсике очень проста. Например, вычисление

2^3 будет: 2**3. Вообще при возведении числа в степень следует помнить, во-первых, что эта операция имеет более высокий приоритет, чем операция умножения, во-вторых, знать следующие правила:

```
A ** Ø = 1
A ** (–B) = 1/(A ** B)
A ** (1/B) = B√A
A ** (B * C) = (A ** B) ** C
A** (–1) = 1/A
A ** (1/2) = SQR A
```

При вычислении тригонометрических функций надо помнить, что

```
SIN (A + 2 * PI) = SIN A
COS (A + 2 * PI) = COS A
```

Случайные числа. Для получения случайных чисел используются функция RND и ключевое слово RANDOMIZE. Например, получение случайного числа в интервале от 1,3 до 2,0 будет 1.3 + 0.7 * RND.

Слово RANDOMIZE используется для установления начала последовательности случайных чисел для функции RND.

Массивы. Оператором DIM формируются массивы чисел, вызываемые из памяти ПЭВМ, специально резервируемой под массив. Переменные в массиве являются его элементами, обладают общим именем и различаются только индексом (номером, записываемым после имени). Элементы массивов называют индексируемыми переменными. Например, имеется список из 10 чисел, описывающих какие-либо объекты. Для размещения данных в компьютер необходимо записать:

```
1Ø FOR N = 1 TO 1Ø
2Ø READ B (N)
3Ø NEXT N
4Ø DATA 1Ø, 2, 5, 17, 16, 12, 20, 8, 17, 3
```

Оператор DIM B(1Ø) определяет массив с именем B и размерностью 1Ø (т.е. 10 индексируемых переменных B (1), B (2), ..., B (1Ø)) и присваивает всем элементам массива значение Ø.

Можно также вводить двухмерные или трехмерные массивы.

Пусть следует сформировать двухмерный массив C с размерностью 3 и 6: DIM C (3,6). Это дает 3*6 = 18 индексируемых переменных:

```
C (1,1),   C (1,2), . . . ,   C (1,6);
C (2,1),   C (2,2), . . . ,   C (2,6);
C (3,1),   C (3,2), . . . ,   C (3,6).
```

Трехмерный массив может формироваться по следующему алгоритму: номер страницы, номер строки, номер столбца Могут быть также строковые массивы. Строки в таких массивах отличаются от скалярных тем, что имеют фиксированную длину, а при-

сваивание им значения осуществляется с усечением справа или добавлением до полной длины пробелами. Имя строкового массива образуется добавлением справа к имени специального символа $ [22].

Логические операции. Числа, строки, выражения могут связываться логическими функциями AND ("И"), OR ("ИЛИ") и NOT ("НЕ"). Функция OR имеет низший приоритет, затем идут функции AND и NOT. Выражение "И" соответствует некоторому выражению истины, если истинны оба этих выражения. Например, X в выражении

IF A$ = ' YES ' AND X > Ø THEN PRINT X

будет напечатано только, когда A$ = 'YES' и X > Ø. Выражение "ИЛИ" соответствует некоторому выражению истины, если истинно хотя бы одно из этих выражений. Выражение "НЕ" (отрицание) истинно, если ложно само выражение и наоборот. Отметим также, что может использоваться условие "< >", которое обратно в логическом смысле условию "=", то есть:

A <> B то же, что и NOT A = B

NOT A <> B то же, что и A = B.

Строки символов. Для определения подстрок (некоторой последовательности символов из строки) существует действие называемое сечением. Общая форма записи

"СТРОКОВОЕ ВЫРАЖЕНИЕ" ("НАЧАЛО" TO "КОНЕЦ").

Если опущено "НАЧАЛО", то по умолчанию подразумевается 1, если – "КОНЕЦ" то подразумевается длина всей строки. Например, следующее выражение истинно:

"ABCDEF" (2 TO 5) = "BCDE".

Следует помнить также такие правила определения подстрок

– "НАЧАЛО" и "КОНЕЦ" должны находиться в пределах строки, иначе будет выдано сообщение об ошибке;

– если "НАЧАЛО" больше, чем "КОНЕЦ", либо обе границы лежат за пределами строки, то результатом будет пустая строка;

– "НАЧАЛО" и "КОНЕЦ" не могут быть отрицательными, иначе выдается сообщение "B INTEGER OUT OF RANGE";

– можно присваивать значения подстроке.

Последнее действие, называемое прокрустианом, обеспечивает следующую особенность присвоения значения подстроке: длинные данные усекаются справа, а короткие дополняются пробелами до длины подстроки.

Можно использовать скобки что позволяет вычислять значение строкового выражения перед тем, как брать сечение. Например:

"ABC" + "DEF" (1 TO 2) = "ABCDE"
("ABC" + "DEF") (1 TO 2) = "AB".

Набор символов. Буквы, цифры, знаки пунктуации обозначаются символами и образуют алфавит (набор символов), используемый компьютером. Отдельные символы, называемые знаками, образуют целые слова (PRINT, STOP и т.п.). ПЭВМ "Спектрум" использует 256 символов с кодами от 0 до 255. Для преобразования из символьной формы во внутреннюю кодовую и наоборот служат две функции CODE и CHR$.

Функция CODE применяется к строке символов и возвращает код внутреннего представления первого символа в строке или 0, если строка пустая.

Функция CHR$ применяется к числу и возвращает один символ, код которого представлен этим числом.

Следующая программа выводит весь отображенный символьный набор:

```
10 FOR A = 32 TO 255 : PRINT CHR$ A; : NEXT A
```

Эти символы, кроме знаков £ (фунт) и c , образуют код ASCII. Первые 15 символов – это черно-белые значки, используемые для изображения рисунков. Их можно ввести с клавиатуры, используя графический режим (при нажатии клавиш "CAPS SHIFT 9" курсор изменится на < G > и цифровые клавиши с 1 по 8 выдают графические символы, обозначенные на клавишах). После графических символов на клавиатуре располагаются символы алфавита от A до U. Графические значения этих клавиш могут определяться самим пользователем, а затем использоваться в графическом режиме.

Цвета. В ПЭВМ "Спектрум" имеется возможность вывода на цветной телевизор восьми цветов (включая черный и белый) и двух уровней яркости [19]. При составлении программы вывода цветов используется кодировка цветов: 0 – черный, 1 – синий, 2 – красный, 3 – фиолетовый, 4 – зеленый, 5 – голубой, 6 – желтый, 7 – белый. Для черно-белого телевизора вышеуказанный ряд модифицируется в переход серых полутонов от черного до белого.

Напомним, что графический экран имеет 768 позиций (24 строки по 32 знакоместа), каждая из которых представляет собой матрицу 8x8 пикселов. Знакоместо отображает цвет тона (оператор INK) и цвет фона (оператор PAPER). Кроме этого, можно сформировать изображения обычной или повышенной яркости, а также мерцающие или немерцающие. Таким образом при формировании цветного изображения следует использовать следующую кодировку:

1) форму символа для знакоместа (8x8 пикселов) определяют чистые и закрашенные точки (0 – белая точка, 1 – черная точка), а цвета тона и фона – операторами INK и PAPER соответственно;

2) цвета фона и тона кодируются от 0 до 7 каждый;

3) яркость: 0 – обычная; 1 – повышенная;

4) мерцание: 0 – постоянно; 1 – мерцание.

Отметим здесь, что для одного знакоместа в 64 пиксела нельз. установить более одного цвета для фона и тона. Это также относится к другим характеристикам изображения (яркость, мерцание). Цвет, яркость и мерцание изображения являются атрибутам знакоместа (а не отдельного пиксела). Изменение этих атрибутов обеспечивают операторы INK, PAPER, BRIGHT, FLASH. Эти операторы имеют следующий формат:

PAPER	– число от 0 до 7;
INK	– число от 0 до 7;
BRIGHT	– число 0 или 1 (0 – выключено; 1 – включено);
FLASH	– число 0 или 1 (0 – выключено; 1 – включено).

В качестве примера ниже приводится программа вывода компьютером на цветной телевизор восьми цветов (включая черный и белый) и двух уровней яркости:

```
10 FOR M=0 TO 1:BRIGHT M
20 FOR N=1 TO 10
30 FOR C=0 TO 7
40 PAPER C:PRINT"    ";:REM 4 COLOURED SPACES
50 NEXT C:NEXT N:NEXT M
60 FOR M=0 TO 1:BRIGHT M:PAPER 7
70 FOR C-0 TO 3
80 INK C:PRINT C;"   ";
90 NEXT C:PAPER 0
1 0 FOR C=4 TO 7
110 INK C:PRINT C;"   ";
120 NEXT C:NEXT M
130 PAPER 7:INK 0:BRIGHT 0
```

Возможно использование больших чисел чем указано выше. Однако это дает другой эффект. Например, число 8 можно использовать во всех четырех операторах позволяющих определить значение ранее установленных атрибутов. Действительно, оператор PAPER 8 не изменит цвета фона (так как такого цвета нет) участк изображения, а поможет выяснить значение предыдущего оператора PAPER. Операторы INK 8, BRIGHT 8, FLASH 8 выдадут значения соответствующих атрибутов. Число 9 может использоваться как средство "контраста" только для операторов INK и PAPER. При формировании цветов операторами INK и PAPER необходимо учитывать следующее: цвета должны быть контрастны друг другу; так, белому цвету подходят темные тона (черный, синий, красный, фиолетовый), а к черному цвету – светлые тона (зеленый, голубой, желтый, белый). Цвет фона будет контрастен цвету тона в каждой выводимой позиции.

Например, при проверке работоспособности компьютера можно запустить программу формирующую на экране дисплея цветные полосы:

INK 9 : PAPER 8 : PRINT AT 0 , 0 ; : FOR N TO 1000 : PRINT N; NEXT N

Отметим здесь, что цветной телевизор, воспринимая три основных цвета: синий, красный и зеленый, остальные цвета формирует путем комбинации основных цветов. Если подать на один участок экрана все восемь цветов, то их все мы не увидим (будет только темное пятно). Однако там где цвета частично накладываются друг на друга, мы увидим цветовую гамму. В качестве примера ниже приводится такая программа (команда INK получена с использованием клавиш "CAPS SHIFT" и "9" в Е-режиме:

```
10 BORDER 0:PAPER 0:INK 7:CLS
20 FOR A=1 TO 6
30 PRINT TAB 6;INK 1;"[][]...[][]":REM 18 INK SQUARES
40 NEXT A
50 LET DATALINE=200
60 GO SUB 1000
70 LET DATALINE=210
80 GO SUB 1000
90 STOP
200 DATA 2,3,7,5,4
210 DATA 2,2,6,4,4
1000 FOR A=1 TO 6
1010 RESTORE DATALINE
1020 FOR B=1 TO 5
1030 READ C:PRINT INK C;"[]...[]";:REM 6 INK SQUARES
1040 NEXT B:PRINT :NEXT A
1050 RETURN
```

Операторы INVERSE и OVER не управляют атрибутами однако определяют способ вывода изображения на экран. В этих операторах используются значения параметров 0 или 1. Например, оператор INVERSE 1 обеспечит вывод символа в негативном (инверсном) изображении (белым по черному фону, вместо черным по белому). Оператор OVER 1 обеспечивает режим расширенного вывода, позволяющий накладывать символы друг на друга, выводить составные символы (стилизованные шрифты). Например, программа для вывода готического шрифта будет

```
10 OVER 1
20 FOR N=1 TO 32
30 PRINT "O";CHR$8;"""";
 0 NEXT N
```

В этой программе управляющий символ CHR$ 8 возвращает на одну позицию.

Операторы INK и PAPER можно использовать как параметры PRINT. Например:

PRINT PAPER 6; "X"; : PRINT "Y".

Таким образом в этой программе только X выводится на желтый фон.

Так как оператор INK (как и другие операторы) не действует

в нижней части экрана, предназначенной для ввода команд и данных, то для изменения цветов здесь служит оператор BORDER "ЦВЕТ".

Изменение цветов можно сделать также с помощью управляющих символов CHR$ 16, ..., CHR$ 21. При этом необходимо учитывать следующее соответствие:

CHR$ 16 – INK;
CHR$ 17 – PAPER;
CHR$ 18 – FLASH;
CHR$ 19 – BRIGHT;
CHR$ 20 – INVERSE;
CHR$ 21 – OVER.

Например, фунция PRINT CHR$ 16 + CHR$ 9 аналогична PRINT INK 9.

Таким образом, для изменения цветов можно пользоваться операторами или соответствующими управляющими символами, которые можно ставить как после номера строки, так и в конце строки.

В расширенном режиме [E] курсора можно пользоваться цифрами, при этом следует помнить следующие рекомендации. Цифры от 0 до 7 устанавливают цвет оператора INK (клавиша "CAPS SHIFT" нажата) и цвет оператора PAPER (клавиша "CAPS SHIFT" не нажата). При нажатии клавиш с соответствующими цифрами будут выведены символы CHR$ 17 и CHR$ КОД ЦВЕТА. Если в это время была нажата клавиша "CAPS SHIFT", то будут выведены символы CHR$ 16 и CHR$ КОД ЦВЕТА. В случае необходимости уничтожения вводимой информации следует дважды нажать клавишу "DELETE". Харктеристики изображения в расширенном режиме [E]:

"8" дает CHR$ 19 и CHR$ 0 – нормальная яркость;
"9" дает CHR$ 19 и CHR$ 1 – повышенная яркость;
"CAPS SHIFT" с "8" дает CHR$ 18 и CHR$ 0 – не мерцающее;
"CAPS SHIFT" с "9" дает CHR$ 18 и CHR$ 1. – мерцающее.

Характеристики изображения в режиме курсора [L]:

"CAPS SHIFT" с "3" дает CHR$ 20 и CHR$ 0 – обычный вывод;
"CAPS SHIFT" с "4" дает CHR$ 20 и CHR$ 1 – инверсный (негативный) вывод.

В табл. 2.5 даны характеристики изображения, формируемого на экране в соответствии с нажатием клавиш "1" ... "0" и режимами курсоров [K], [L], [C], [G], [E].

Рассмотрим функцию ATTR Она имеет следующий формат:

ATTR ("СТРОКА", "СТОЛБЕЦ")

Значения параметров функции аналогичны значению параметров функции AT. Результат (значения атрибутов для соответствующей позиции экрана) представляет собой сумму четырех чисел:

1) 128 – знакоместо мерцающее; 0 – знакоместо обычное;
2) 64 – повышенная яркость, 0 – обычная яркость;
3) 8*(КОД ЦВЕТА ФОНА);
4) КОД ЦВЕТА ТОНА.

Таблица 2.5

Управление экраном

РЕЖИМ КУРСОРА							
[K], [L], [C]			[G]		[E]		
ИСПОЛЬЗОВАНИЕ (НАЖАТИЕ) КЛАВИШ							
одна клавиша	вместе с SYMBOL	вместе с CAPS	одна клавиша	вместе с любой	одна клавиша	вместе с CAPS	вместе с SYMBOL
1	2	3	4	5	6	7	8
ОТОБРАЖАЕМАЯ ИНФОРМАЦИЯ НА ЭКРАНЕ							
1	!	EDIT	**** ****	**** **** ******* *******	Фон голубой	Тон голубой	DEF FN
2	@	CAPS LOCK	**** ****	**** ******** ******** ********	Фон красный	Тон красный	FN
3	#	TRUE VIDEO	******** ********	******** ********	Фон фиолетовый	Тон фиолетовый	LINE
4	$	INVERS VIDEO	**** ****	******** ******** **** ****	Фон зеленый	Тон зеленый	OPEN

Окончание табл. 2.5

1	2	3	4	5	6	7	8
5	%	Курсор ВЛЕВО	**** **** **** ****	**** **** **** ****	Фон синий	Тон синий	CLOSE
6	&	Курсор ВНИЗ	**** **** **** ****	**** **** **** ****	Фон желтый	Тон желтый	MOVE
7	,	Курсор ВВЕРХ	******** ******** **** ****	**** ****	Фон белый	Тон белый	ERASE
8	(	Курсор ВПРАВО		******** ******** ******** ********	Нормальная яркость	Нормальное мерцание	POINT
9	(	Графический режим	Графический выход	Графический выход	Повышенная яркость	С мерцанием	CAT
0	–	DELETE	DELETE	DELETE	Фон черный	Закрашенный черный	FORMAT

Например, запись строки следующего содержания: знакоместо мерцающее, обычной яркости, желтый фон, синий тон

$$128 + 0 + (8 * 6) + 1 = 177$$

В качестве примера приведем также разбивку экрана под шахматную доску.

```
5 REM DRAW BLANKBOARD
10 LET BB=1:LET BW=2:REM RED AND BLUE FOR BOARD
15 PAPER BW:INK BB:CLS
20 PLOT 79,128:REM BORDER
30 DRAW 65,0:DRAW 0,-65
40 DRAW -65,0:DRAW 0,65
50 PAPER BB
60 REM BOARD
70 FOR N=0 TO 3:FOR M=0 TO 3
80 PRINT AT 6+2*N,11+2*M;" "
90 PRINT AT 7+2*N,10+2*M;" "
100 NEXT M:NEXT N
110 PAPER 8
120 LET PW=6:LET PB=5:REM COLOURS OF WHITE AND BLACK
                                    PIECES
200 DIM B*(8,8):REM POSITIONS OF PIECES
205 REM SET AP INITIAL POSITION
210 LET B*(1)="RNBQKBNR":REM LITTLE LINE 240
220 LET B*(2)="PPPPPPPP":REM LITTLE LINE 230
230 LET B*(7)="PPPPPPPP":REM BIG LINE 220
240 LET B*(8)="RNBQKBNR":REM BIG LINE 210
300 REM DISPLAY BOARD
310 FOR N=1 TO 8:FOR M=1 TO 8
320 LET BC=CODE B*(N,M): INK PW
325 IF BC=CODE " " THEN GO TO 350:REM SPACE
330 IF BC>CODE  Z" THEN INK PB:LET BC=BC-32:REM LOWER
    CASE FOR BLACK
340 LET BC=BC+79:REM CONVERT TO GRAPHICS
350 PRINT AT 5+N,9+M;CHR*BC
360 NEXT M:NEXT N
400 PAPER 7:INK 0
```

Операции с экраном (графика). Напомним, что для отображения графической информации на экране телевизора используются 22x32 = 704 символьные позиции, каждая из которых представляется квадратом 8x8 точек (пикселов). Пиксел задается двумя числами. Первое число задает координату X (удаление, в пикселах, до левой границы экрана), а второе число – координату У. В компьютере для отображения информации на экране телевизора используются операторы PLOT, DRAW, CIRCLE, POINT.

Оператор PLOT X,Y отображает в цвете (INK) пиксел с указанными координатами. Например, программа, вычерчивающая график функции sin X (X в интервале от 0 до 2π), будет

```
10 FOR N=0 TO 255
20 PLOT N,88+80*SIN(N/128*PI)
30 NEXT N
```

С помощью оператора DRAW можно чертить линию, заданную в форме

DRAW X, Y –

Началом этой линии является точка текущей PLOT-позиции, на которой завершился один из предыдущих операторов PLOT, DRAW или CIRCLE. То есть оператор DRAW задает длину и направление вычерчивания линии но не начальную точку. Заметим, что установка начальной точки в левый нижний угол экрана производится операторами RUN, CLEAR, CLS и NEW.

Чертить линии можно в цвете, устанавливаемом для целой позиции (а не для отдельного пиксела).

С помощью оператора DRAW можно также вычертить отрезок дуги используя для этого дополнительное число, задающее угол этой дуги (в радианах)

DRAW X, Y, A

Здесь, если A положительно, то дуга вычерчивается влево, а если отрицательно – вправо. При A = 2π сформируется полная окружность. Например, если задать начальную точку (50, 50) и конечную точку (75, 75), то программа вычерчивания полуокружности будет

10 PLOT 50 , 50 : DRAW 25, 25, PI

Оператор CIRCLE вычерчивает полный круг, задаваемый координатами его центра и радиусом (CIRCLE X,Y РАДИУС).

Далее рассмотрим операции с экраном, созданным в графическом редакторе [22]. Удобным является графический редактор ARTSTUDIO. Изображение, созданное в редакторе, можно выгрузить на ленту магнитофона в виде блока кодов (длиной 6912 байтов), а затем по мере необходимости загрузить командой

LOAD "" CODE 16384, 6912

По этой команде можно сразу загрузить изображение на экран. Но хранить его там нельзя. Поэтому изображение можно загрузить в произвольное место памяти и вызывать его оттуда на экран с помощью программы на Бейсике.

Однако из-за ограниченных возможностей языка Бейсик изображение на экране сформируется более чем за 20 с. Для быстрой записи изображения на экране (около 1 с) необходимо выполнить программирование в машинном коде. В микропроцессоре Z80 для перемещения блоков в машинном коде имеется команда LDIR (ее код 237,176).

Ниже приводится программа для вывода на экран картинки, хранящейся по адресу 30000 (код 48,117). Эта программа вызывается командой

RANDOMIZE USR 65000 (65000 – адрес программы):

```
10 FOR I=1 TO 12
20 READ A
30 POKE 59999+I,A
40 NEXT I
50 DATA 33,48,117,1,0,27,17,0,64,237,176,201
```

Если в компьютере хранится несколько изображений, то загрузка нужного, например второго изображения, хранящегося по адресу 37000 (код 136,144), осуществится следующим образом: необходимо выполнить POKE 65001,136 и POKE 65002,144, а затем дать команду

RANDOMIZE USR 65000.

С целью экономии памяти компьютера используются операции "сжатия" (компрессирования) изображения. В качестве обратной операции используется процедура декомпрессирования изображения.

Указания. Для реализации различных остановок выполнения программы используются операторы PAUSE, PEEK и INKEY$ [22]

В случае необходимости задержки выполнения программы в течение N телевизионных кадров используется оператор PAUSE N (N = 0÷65535). Число 65535 соответствует 22 мин, а число 0 – неограниченно большому времени. После остановки программы работа может быть возобновлена до окончания времени, определенного в операторе PAUSE путем нажатия любой клавиши(кроме клавиши "CAPS SHIFT", вызывающей прерывание).

Оператор PEEK используется в программах моделирования часов (применяемых при различных остановках). Например, числа PEEK 23674, PEEK 23673 и PEEK 23672 выделяют адреса ячеек памяти компьютера, с помощью которых происходит подсчет 1/50 долей секунды. В каждой из этих ячеек подсчитывается сумма от 0 до 255. После достижения числа 255 в любой из ячеек она сбрасывается в 0.

Оператор INKEY$ (без аргументов) считывает с клавиатуры. Например, при нажатии какой-либо клавиши результатом будет символ, соответствующий этой клавише в режиме маркера <L> или пустая строка.

Программирование звуков. Компьютер способен воспроизв дить звуки с помощью оператора BEEP, запись которого производится следующим образом:

BEEP N, M

Здесь N (продолжительность звука) и M (высота звука) – некоторые числовые выражения. Число N задается в секундах, а высота – в полутонах от основного тона ноты "до" : при положительных числах – выше ноты "до", а при отрицательных – ниже ноты "до" На рис. 2.5 приведены значения нот одной октавы, привязанные к клавиатуре фортепиано. Как видно из рисунка, для получения более высоких или более низких нот следует прибавить или отнять 12 для каждой октавы вверх или вниз.

Компьютер может менять длительность и высоту звука (путе введения переменной KEY). Однако компьютер не способен одн временно исполнять несколько нот, что не позволяет воспроизводит ему сложные мелодии.

В заключение приведем несколько программ, выполняющих

простую мелодию. Например, фрагмент программы, исполняюще˘ последовательно ноты до предельно высокой, будет

FOR N = 0 TO 1000 : BEEP . 5 , NEXT N

Эта программа завершится сообщением об ошибке "B".

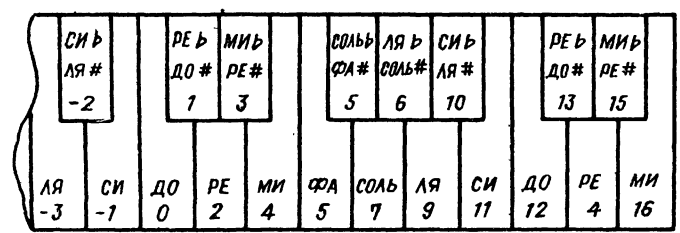


Рис. 2.5. Ноты одной октавы и их соответствие числовым выражениям

Программа, исполняющая гамму до–минор, использует чистые ноты от среднего "до" до верхнего "до":

```
10 BEEP :5,0:BEEP .5,2,BEEP .5,4:BEEP .5,5:BEEP .5,7:
   BEEP .5,9:BEEP .5,11:BEEP .5,12:STOP
```

Эту мелодию можно сделать более естественной следующим образом:

```
20 BEEP .5,0:BEEP .5,2.039:BEEP .5,3.86:BEEP .5,4.98:
   BEEP .5,7.02:BEEP..5,8.84:BEEP .5,10.88:BEEP .5,10.88
   BEEP .5,12:STOP
```

Можно сделать клавиатуру компьютера клавишами музыкального инструмента, выполнив переключение

POKE 23609,255

Здесь число 255 определяет продолжительность нахождения компьютера в этом состоянии.

Музыку можно также вывести на внешние устройства, подключаемые к выходным разъемам "MIC" и "EAR" [22].

2.3.4. Методы защиты программ

Защита программ состоит в исключении возможности просмотра текста программы и внесения в него изменений пользователем, не уполномоченным на это автором программы [25].

Методов защиты программы существует очень много, но мы рассмотрим только основные принципы. Можно выделить три направления.

1. Исключить возможность остановки (прерывания) работы программы.

2. Исключить возможность подачи команды LIST для распечатки текста программы.

3. Сделать листинг программы нечитаемым.

Исключение возможности остановки программы

Прежде всего необходимо сделать программу автостартующей со строки N. Это осуществляется при выгрузке программы командой SAVE "имя" LINE N. Теперь программа после загрузки будет сама стартовать с указанной строки N. Однако она может быть остановлена командой BREAK или по сообщению об ошибке.

Отключить клавишу "BREAK" можно, если в строке, с которой происходит автостарт, поместить команду

POKE 23613, PEEK 23730–5.

Если программа хорошо отлажена, то остановку по ошибке можно исключить. Необходимо иметь в виду, что чаще всего при вскрытии программы сознательно вносят ошибку во время исполнения оператора INPUT. Например, если программа требует от пользователя ввести какое-либо число, он сознательно нажимает на клавишу с буквой. Программа останавливается с сообщением VARIABLE NOT FOUND. Поэтому не рекомендуется применять оператор INPUT, а организовать опрос клавиатуры на основе функции INKEY$.

Если в строке автостарта поместить нижеприведенные команды, то при попытке сделать BREAK или при появлении ошибки программа будет сбрасываться

LET ERR = 256 * PEEK 23614 + PEEK 23613 : POKE ERR + 1,0

Примерно тот же эффект дает помещение в стартовой строке команды POKE 23659,0 .

Одним из способов остановки автостартующей программы является использование при загрузке команды MERGE "имя" вместо LOAD "имя". Если ваша программа имеет размер больше 7K, то защитить ее от этого приема можно, если перед выгрузкой подать следующие прямые команды:

LET X = 256 * PEEK 23636 + PEEK 23635 : POKE X, 60 : POKE X + 1, 0

Другой способ защиты от MERGE: перед выгрузкой программы на ленту дать ложную информацию о том, что якобы, первая строка вашей программы имеет чрезмерно большую длину. Это делается прямой командой

POKE (PEEK 23635 + 256 * PEEK 23636 + 3), 255

Отключение команды LIST

Для достижения этой цели первой строкой программы нужно ввести 1 РЕМ, а перед выгрузкой программы на ленту дать прямую команду:

POKE (PEEK 23635 + 256 * PEEK 23636), 100.

Теперь программу можно запускать (по команде RUN), но читать нельзя.

Как сделать текст программы нечитаемым

Простейший прием, применяемый во многих программах,

состоит в том, что устанавливают одинаковыми цвета символов (INK) и фона (PAPER). Например: 10 INK 7 : PAPER 7.

В тех местах, где программа должна сделать вывод на экран, в операторе PRINT вставляют в качестве временных правильные цвета

20 PRINT INK 0 ;"сообщение".

Другой способ состоит в искажении набора символов путем задания "фальшивого" значения системной переменной CHARS. Например:

POKE 23606,8 : PRINT "сообщение":POKE 23606,0.

Если же адресовать системную переменную CHARS в те области памяти, где вообще ничего нет, например POKE 23607,200, то все символы будут выглядеть как пробелы, и на экране вообще ничего не будет. Но надо не забывать перед всяким оператором PRINT включать истинное значение CHARS и выключать его после PRINT.

Еще один регулярно используемый прием состоит в организации "нулевой" строки, которая реально исполняется программой, но не может быть удалена при редактировании. Чтобы первая строка стала нулевой, достаточно дать команду: POKE 23755,0: POKE 23756,0.

2.4. Работа с компьютером

2.4.1. Разъемы

На фирменных компьютерах "Спектрум-48" ("ZX-Spectrum-48") и "Спектрум+" ("ZX-Spectrum+") расположены на задней панели следующие разъемы [15, 22, 23, 24]:

EAR – для подключения магнитофона при загрузке программ;
MIC – для подключения магнитофона при выгрузке программ;
TV – для подключения телевизора через антенный вход;
9VDC – для подключения внешнего блока питания.

Кроме того, там же расположен 54-контактный краевой разъем для подключения периферийных устройств. Если у вашей ПЭВМ нет периферии, то этот разъем следует закрыть (например, заклеить липкой лентой), так как случайное попадание металлического предмета на контактные дорожки во время работы компьютера может привести к выходу его из строя. Нельзя подключать периферийные устройства к работающему компьютеру, так как все шины у ПЭВМ "Спектрум" небуферизированы.

В компьютерах "Спектрум+128" и "Спектрум+2" кроме антенного выхода имеется также и выход RGB, который служит для подключения к RGB-монитору или на выход цветовых усилителей телевизора.

Особенности же большинства самодельных компьютеров

состоят в том, что они не имеют 54-контактного краевого разъема (ввиду отсутствия и дороговизны периферийных устройств), разъемы EAR и MIC у них совмещены в один (обычно для этих целей применяют обыкновенные бытовые 5-штырьковые разъемы) а на телевизор выводятся сигналы R, G, B и синхросмесь. Кроме того предусматривают разъем и для подключения джойстика (обычно KEMPSTON).

2.4.2. Порты

Компьютер может считывать некоторую информацию и записывать ее в свою оперативную память по командам PEEK и POKE. Вся память компьютера, и ПЗУ, и ОЗУ, представляется совокупностью адресов от 0 до 65535, каждый из которых адресует один байт.

Таким же образом можно адресовать и еще 65536 адресов, называемых портами ввода–вывода. Они используются процессором для связи с клавиатурой и принтером и могут управляться операторами Бейсика IN и OUT.

Оператор IN аналогичен оператору PEEK : IN ADDR. Он использует один аргумент – адрес порта, и позволяет считать один байт из указанного порта.

Оператор OUT подобен оператору POKE : OUT ADDR VALUE. Он записывает указанные данные (VALUE) в заданный порт вывода.

ПЭВМ "Спектрум" оперирует с шестнадцатью разрядными адресами, которые обычно обозначаются буквой A : A15, A14, A13, ..., A1, A0 .

Младшие биты адреса A0 , A1, A2, A3, и A4 имеют очень важное значение. Как правило, они установлены в 1. Но если хотя бы один из них в 0 , то это предписывает компьютеру некоторые действия. Не более чем один из этих пяти битов может быть установлен в 0.

Биты A6 и A7 игнорируются, так что, если пользователь знаком с электроникой, то может использовать их по своему усмотрению.

Биты A8, A9 используются иногда для получения дополнительной информации.

Информационный байт (данные) обозначим буквой D : D7, D6, D5, ..., D1, D0 .

Теперь представим список адресов портов. Имеется целый ряд входных адресов для чтения с клавиатуры, а также магнитофона (разъем EAR). Сама клавиатура разбита на 8 полурядов по 5 клавиш в ряду:

IN	65 278	считывает ряд от	"CAPS SHIFT"	до	"V",
IN	65022	считывает ряд от	"A"	до	"G",
IN	64 510	считывает ряд от	"Q"	до	"T",
IN	63 486	считывает ряд от	"1"	до	"5",

IN 61 438 считывает ряд от "0" до "6",
IN 57 342 считывает ряд от "P" до "Y",
IN 49 150 считывает ряд от "ENTER" до "H",
IN 32 766 считывает ряд от "SPACE" до "B".

Эти адреса могут быть вычислены из выражения [15, 22]: 254 + 256 * (255 – 2 ** N), при N, пробегающем от 0 до 7.

В байте, считанном с клавиатуры, биты от D0 до D4 служат для обозначения пяти клавиш в данном полуряду. Причем бит D0 – для крайней клавиши, а бит D4 – для той, что ближе к центру. Если один из этих битов находится в 0, это значит, что соответствующая ему клавиша нажата. Бит D6 изменяет свое значение при чтении с разъема EAR.

Выходной порт 254 отведен под громкоговоритель (D4) и разъем MIC (D3), а также установку цвета (D2, D1, D0).

Порт 251 обеспечивает связь с принтером как для чтения, так и для записи (чтение – для проверки готовности принтера к работе)

Порты 254, 247 и 239 используются для связи с дополнительными устройствами (микродрайв).

2.4.3. Принтеры и их подключение к компьютеру

Наиболее желаемое периферийное устройство, о подключении которого мечтает большинство пользователей, – это принтер. Отметим, что к компьютеру можно подключить практически любой принтер, но при этом могут возникать те или иные трудности.

Прежде всего надо определить для каких целей он нужен. Если основное направление пользователя – программирование, то может подойти небольшой узкопечатный "ZX-printer", который подключается к краевому разъему компьютера без каких-либо интерфейсов и не требует никакого программного обеспечения для работы (все уже содержится в ПЗУ компьютера). Такой принтер имеет ширину бумажной ленты 10...13 см и выводит 32 символа в строке. Он также может выполнять графическую копию экрана. Хотя качество печати далеко не идеальное, такие устройства сравнительно дешевы, просты и удобны в работе и, если их применять в основном для распечатки текстов программ или результатов расчетов, они вполне соответствуют поставленным задачам.

Наиболее распространены 4 модели таких принтеров [15, 22, 23]. Это "ZX-printer", "Alphacom 32", "Timex-2040", "Seikosha GP 50 S".

"ZX-printer"– стандартное устройство. Печать выполняется на алюминизированной рулонной бумаге. Качество печати наихудшее из этих 4 моделей.

"Alphacom 32" и "Timex 2040" – практически идентичны. Они работают со специальной термочувствительной бумагой и получили широкое распространение.

"Seikosha GP 50 S" – выпускался в Японии. Он имеет эстетичный дизайн и наилучшее качество печати. Бумага – обычная рулонная, шириной 13 см. Применяется узкая красящая лента.

Если необходимо получение качественных документов на листах стандартного формата, то следует использовать полноформатный принтер, выпускаемый двух типов: точечно-матричные принтеры и принтеры типа "ромашка". Отличие этих принтеров (друг от друга) заключается в конструкции печатающего элемента. В первом случае – это набор иголок, оставляющих через красящую ленту мозаичный отпечаток на бумаге. Из точек складывается изображение символа. В принтерах второго типа печатающие литеры закреплены на вращающемся прорезном диске ("ромашке"). Этот диск может быть сменным для замены набора литер.

Матричные принтеры значительно быстрее работают, могут гибко (программно) перестраиваться с одного вида шрифта на другой, но по качеству печати только самые лучшие из них могут приближаться к принтерам типа "ромашка".

В основном принтеры "ромашка" применяются в тех случаях, когда имеется необходимость в большом количестве деловой переписки и качество напечатанного документа свидетельствует об уровне представительства. Надо также отметить, что выдача графических изображений на принтерах этого типа исключена.

Наиболее гибким и многофункциональным устройством является точечно-матричный принтер. Остановимся на этих устройствах и отметим, какие вопросы нужно для себя решить, прежде чем приобретать такой принтер.

Тип используемого интерфейса. Существуют два основных метода обмена информацией между компьютерами и периферийными устройствами – параллельный и последовательный. При параллельном подключении все биты, составляющие байт посылки, передаются одновременно по параллельным шинам данных. При последовательном – байт передается бит за битом по очереди. Исторически за интерфейсами, реализующими параллельный метод передачи, закрепилось название "Центроникс", а последовательный – RS-232.

Большинство современных компьютеров имеют встроенный интерфейсный порт того или иного типа (а нередко и обоих). Однако ПЭВМ "Спектрум-48" его не имеет. Наибольшее распространение получили следующие модели интерфейсов [22, 23]:

"Интерфейс-1" – имеет последовательный порт RS-232, к которому может подключаться принтер, имеющий такой же интерфейс. Никакой программной поддержки этот интерфейс не требует, она находится в его ПЗУ.

"ZX-Lprint" – этот интерфейс может настраиваться и поддерживает аппаратуру как с интерфейсом "Центроникс" так и с RS-232.

"Tasprint" – интерфейс типа "Центроникс". Требует для своей работы специальной программы (драйвера).

"Kempston-E" – интерфейс типа "Центроникс", также требует драйвера.

Компьютеры "Спектрум+128" и "Спектрум+2" имеют уже встроенный порт RS-232. Зато они не могут работать с ZX-принтерами, так как область буфера принтера там занята дополнительными системными переменными. И хотя существуют программные пути устранения этого недостатка, но использовать этот порт затруднительно, так как он поддерживается только в режиме 128 К, а большинство прикладных программ (текстовые редакторы, графические редакторы и т. д.) работают в режиме 48 К.

Рассмотрим основные характеристики и режимы работы принтеров.

Наличие дополнительного набора знаков. Далеко не все импортные принтеры имеют встроенный русский набор знаков. Наилучшие модели имеют возможность загрузки шрифта от компьютера.

Применяемая бумага. Некоторые принтеры используют термобумагу, что создает ощутимые неудобства в связи с отсутствием ее в широкой продаже. Кроме того, качество печати в этом случае не очень высокое.

Красящая лента. Многие принтеры работают с лентой, находящейся в специальных неразборных кассетах Это удобно при смене ленты, но может поставить пользователя в трудное положение, если доступа к таким кассетам у него нет, а применение обычной ленты на катушках невозможно.

Скорость печати. Если предполагается исполнение многочисленных копий документов, то этот параметр может быть важным. Обычно точечно-матричный принтер имеет скрость печати 80...100 зн/с. По этому параметру можно судить о совершенстве той или иной модели принтера. Наилучшие из них имеют скорость печати свыше 180 зн/с.

Количество иголок в печатающей головке. Как правило, принтер среднего качества имеет 9 иголок в печатающей головке. Наиболее совершенные модели имеют до 24 иголок.

Наличие режима NLQ. Режим NLQ (NORMAL LETTER QUALITE) обеспечивает печать символами, по качеству приближающимися к стандартной пищущей машинке. Такой режим имеется далеко не у всех принтеров. Печать в этом режиме выполняется значительно медленнее, зато он позволяет получать документы высокого качества.

Коды управления принтером. Принтер может не только распечатывать текст, который поступает в него из компьютера, но и выполнять дополнительно целый ряд специфических действий (подчеркивать текст, переключаться с одного набора знаков на

другой изменять размер шрифта, выполнять печать двойным ударом и еще многие другие действия). Эти режимы переключаются с помощью управляющих кодов, которые в нужных местах вставляются в текст. Как правило, управляющие коды состоят из последовательности кодов, начинающихся с символа 27 (ESCAPE), поэтому часто коды управления принтером называют ESCAPE-кодами. В зависимости от того, какому коду какое действие принтера соответствует, различают несколько стандартов этих кодов. Наиболее широко распространена система кодов фирмы Epson . В зависимости от того, поддерживает ли принтер этот стандарт кодов, он может быть или не быть Epson-совместимым. Желательно, чтобы такая совместимость была, тогда можно распечатывать подготовленные документы не только на своем принтере: это также упростит задачу настройки фирменных программ на работу с принтером, так как они, как правило, готовятся в расчете на Epson-совместимые принтеры.

DIP-переключатели. Эти переключатели служат для задания определенных режимов работы принтера. Многие их функции дублируются и могут быть изменены кодами управления принтером (о которых уже говорилось выше), но те условия, которые заданы этими переключателями, являются исходными в момент включения принтера. Чем их больше, тем больше возможностей имеет принтер. Нормально их должно быть порядка 24 штук.

Буфер принтера. Принтер должен иметь определенный объем свободной оперативной памяти, где он размещает информацию, поступающую от компьютера, и выполняет над ней преобразования перед выводом на печать. Именно наличие буфера позволяет принтеру, например, печатать при обратном проходе головки (справа налево), что вдвое повышает его быстродействие. При наличии достаточно емкого буфера компьютер очень быстро может перебросить в него выводимую информацию, и пользователь может, например, готовить к печати очередной блок, пока принтер ведет печать. Если же емкость буфера мала, компьютеру приходится все время подзагружать буфер, и для других целей его использовать уже нельзя. Для большинства практических задач емкость буфера порядка 0,5 К достаточна, но лучше, если она больше 2 К, и наилучшие модели принтеров имеют емкость буфера больше 4 К.

Выбор текстового редактора. Текстовые редакторы предоставляют наиболее удобные условия для работы с принтером, поэтому, предполагая подключение принтера, надо иметь в виду и необходимость наличия редактора. Наиболее широко распространены различные версии текстового редактора TASWORD [15, 22]. Эта программа отличается простотой в работе, предоставляет возможность изображения на экране как 32 символов в строке, так и 64 Имеет несколько кодов управления принтером, которые вставляются в

текст в виде символов блочной графики. Достаточно просто переделывается на работу с русским или другим шрифтом.

Наиболее совершенным и гибким является редактор THE LAST WORD 2. Среди программ для бытовых компьютеров любых моделей трудно найти сравнимый с ним по возможности. Он может воспроизводить текст на экране в режимах 40,48,60 и 80 знаков в строке. Имеет очень удобное меню настройки условий печати, обширный набор операторов управления принтером, который к тому же может гибко перестраиваться без выхода из программы. Имеет встроенный таймер и калькулятор, средства поиска информации и мощные средства для оформления текста и составления различных документов. Также к нему прилагаются дополнительные файлы (пакет программ), позволяющие перенастроить редактор на работу в дисковых операционных системах "TR DOS", "OPUS", "GORDON" и др. Есть у него и возможность работы с текстами, выполненными в других редакторах и базах данных.

Правда, эта программа значительно сложнее в освоении по сравнению с редакторами TASWORD и имеет достаточно объемную инструкцию по работе.

2.4.4. Работа с магнитофоном

Подключение к магнитофону [15, 22]. Как уже говорилось выше, для связи с магнитофоном фирменные компьютеры имеют два разъема EAR (для загрузки) и MIC (для выгрузки). Причем гнездо MIC компьютера соединяется с микрофонным входом магнитофона, а гнездо EAR – с выходом магнитофона, предназначенным для подключения наушников. Здесь следует отметить, что при выгрузке программы на ленту загрузочный шнур EAR должен быть обязательно отключен, в противном случае произойдет закольцовывание сигнала и запись будет испорчена. Это ограничение не распространяется на большинство самодельных компьютеров, у которых цепи EAR и MIC электрически разорваны. Кроме того, для связи с магнитофоном у них обычно предусмотрен только один разъем, и поэтому приходится применять специальный шнур. Но зато те пользователи, которые работают с самодельными компьютерами, могут загружать программы непосредственно с линейного выхода магнитофона. Если копии качественные, то проблем при их загрузке не возникает. Если же качество копий недостаточно хорошее, то можно поэкспериментировать с уровнем громкости и регулировкой тембра, для чего необходимо подключиться к выходу магнитофона, предназначенному для подключения наушников

В общем случае загрузка программ выполняется по команде LOAD "" <ENTER> (между кавычками пробела быть не должно). Основную проблему при этом представляет плохая загружаемость

программ, о чем свидетельствует сообщение об ошибке TAPE LOADING ERROR, "зависание" программы после загрузки или самопроизвольный сброс. Причина не обязательно связана с плохой загрузкой. Ниже приводятся некоторые полезные советы, которыми надо руководствоваться, чтобы улучшить загружаемость программ.

1. Надежную загрузку может обеспечить только высококачественная лента, если она записана на том же магнитофоне, на котором и воспроизводится.

2. Для работы с компьютером годится любой бытовой магнитофон, причем чем он проще, тем легче добиться надежной работы. В то же время желательно, чтобы он имел регуляторы тембров (по возможности раздельные). Очень шорошо, если есть регулировка скорости движения ленты, так как компьютер достаточно "чувствителен" к отклонению скорости. Монофонические магнитофоны во всех случаях предпочтительнее, чем стерео. А для того чтобы быстро находить нужную программу, необходим счетчик ленты.

3. При загрузке с выхода, предназначенного для подключения наушников, уровень громкости следует установить примерно на 70–80 % от максимального, и, если загрузка не идет, следует изменять уровень громкости. Необходимо помнить, что программы чаще страдают от избыточного уровня, чем от недостаточного. Поэтому следует вести поиск в сторону уменьшения уровня.

4. Установка регуляторов тембра также имеет большое значение Нормально "высокие" надо ставить на 80 % от максимума, а "низкие" – примерно на 20 %. Если программа решительно не загружается, а регулировка громкости ничего не дает, то необходимо эти регуляторы установить так: 50–50 % и 20–80 %.

5. Загрузка программ сопровождается перемещением полос по бордюрному полю экрана телевизора. По анализу этого изображения можно также сказать о настройке магнитофона. Сначала идут в течение 3–5 с широкие красные и голубые полосы. Это так называемый пилоттон. Он нужен для того, чтобы процессор подготовился к загрузке. Ширина красных и голубых полос должна быть одинаковой. Если голубые полосы шире, то, значит, уровень слишком большой. Если шире красные полосы, – слишком низкий. Эти полосы должны медленно перемещаться по экрану. Слишком быстрое их перемещение говорит о несоответствии скоростей используемого магнитофона и того, на котором была сделана запись. Неравномерное перемещение (с ускорением и рывками) свидетельствует о некачественности лентопротяжной системы магнитофона (используемого или того, на котором делалась запись).

Бывают случаи, когда широкие красные и голубые полосы "превращаются" в широкие желтые и синие. Это означает, что следует, видимо, поменять местами установку регуляторов тембра.

6. Если у пользователя стереомагнитофон и программа не загружается, следует загружать ее отдельно по левому и правому каналам.

7. Если все пути регулировки исчерпаны, но программа не загружается, и при этом точно известно, что копия работоспособна, то, по-видимому, причина в настройке головок у писавшего и воспроизводящего магнитофона. Как правило, бытовые магнитофоны имеют в крышке отверстие для узкой отвертки, чтобы можно было при воспроизведении записи на слух поворотом подпружиненного винта регулировать положение головки. При этом следует добиваться наиболее четкого и резкого звучания сигналов. Обычно бывает достаточно поворота в ту или иную сторону на угол 120 градусов. Если имеется только один магнитофон и он используется также для записи программ, то не следует регулировать головку, так как придется периодически сталкиваться с необходимостью юстировки. Самый лучший вариант – использовать для воспроизведения программ, полученных со стороны, отдельный магнитофон, наиболее простой и дешевый, а для записи программ – наиболее качественный специализированный аппарат.

8. Обычно программы состоят из нескольких блоков, которые загружаются последовательно один за другим. Иногда бывает очень трудно загрузить программу, разные блоки которой требуют разной установки органов управления магнитофона. Для того чтобы обойти эту проблему, надо проводить загрузку в копирующую программу ("копировщик"). Если, предположим, при загрузке второго блока произойдет сбой, то первый блок уже загружен, так что перезагружать его не надо. К тому же по окончании длительной трудоемкой загрузки можно сразу же выполнить качественную копию и больше не иметь с этой программой проблем.

9. Сделать процесс настройки магнитофона наглядным и более простым можно с помощью специальных программ, например с помощью программы TAPER.

Программа TAPER. Предназначена для настройки магнитофона. Она анализирует (раскладывает по частотам) сигнал, поступающий на вход EAR компьютера, и изображает его на экране в виде спектра. Вертикальная шкала имеет три метки. Против этих меток должны находиться пики сигналов во время загрузки. Самая верхняя метка указывает на пик во время считывания пилоттона, а две другие – уровень "нулей" и уровень "единиц". При прослушивании цифровой записи через программу TAPER пики в участках, отмеченных метками, должны быть ярко выражены и никаких побочных всплесков не должно быть.

Копирование программ. Если программа написана пользователем самостоятельно на Бейсике, то выполнить ее копию несложно. Для этого достаточно набрать команду SAVE "имя" и нажать клавишу "ENTER" Чтобы программа автостартовала после загрузки, начиная со строки M команда должна иметь вид: SAVE "имя" LINE M. Если в программе есть блоки, написанные в машинном коде, то нужно

дать команду SAVE "имя" CODE M, N, где M – адрес, с которого начинается блок кодов, а N – длина этого блока в байтах. При загрузке программ блоки, написанные на Бейсике, индицируются на экране как PROGRAM ..., а блоки машинных кодов – как BYTES... .

Не вызывает затруднений выгрузка программы с известными адресами, в которых располагаются блоки. Трудности могут возникнуть при работе (загрузка, выгрузка) с фирменной программой, когда пользователю не известны адреса блоков, из которых она состоит. Кроме того, она может автостартовать, а остановить ее сложно.

Упростить процесс копирования и автоматизировать его можно с помощью специальных копирующих программ. Эти программы имеют свою нестандартную загрузочную систему. Рассмотрим несколько наиболее широко распространенных программ-копировщиков [22]. COPY 86/M – это, по-видимому, наиболее удобный и наглядный копировщик общего применения. В исходном состоянии он имеет объем свободного пространства 45000 байтов и способен компрессировать данные при загрузке. Компрессирование состоит в том, что, например, вместо последовательности из N нулевых байтов записывается один "нуль", а следующий байт указывает их количество (N). Это позволяет загружать в копировщик значительно больше 45 K. Особенно много места экономится при загрузке экранов, так как в графических изображениях часто встречаются длинные последовательности нулей.

Результат компрессии данных отправляется на хранение в область экрана, что приводит к появлению на экране полос и точек напоминающих телеграфный код.

Перечень команд программы COPY 86/M:

L – загрузить файлы;

C – выгрузить (скопировать) файлы; пауза между записями равна 1,5 с;

M – то же, но с паузой 3 с;

V–VERIFY – проверка выгруженных записей;

D–DELETE – удаление ненужных записей;

H – перевод в шестнадцатиричную систему;

X – просмотр записей с отменой ранее поданных команд,

S – то же, но без отмены команд;

B – просмотр Бейсик-программы;

A–ALL – после этой команды очередная команда будет одновременно распространяться на все записи;

BREAK – переход в исходное положение

Все команды подтверждаются нажатием клавиши "ENTER".

На экране файлы записываются в "окне", имеющем 4 строки. Если записей больше чем 4, то одновременно вывести их на экран нельзя, поэтому и необходимы клавиши "S" и "X" для "прокручивания" записей через "окно".

К сожалению, эта программа не работает со многими самодельными компьютерами. В таких случаях, как правило, пользуются другой компрессирующей программой TF COPY.

TF COPY – после загрузки программы на экране появляется исходное меню. Если нажать клавишу "Ø ", то программа стартует и на экране появится основное меню программы:

LOAD SAVE DELETE VERIFY MODE

Для выбора нужного режима необходимо нажать первую букву нужной команды:

LOAD – загрузка файлов;
SAVE – выгрузка файлов;
DELETE – удаление файлов;
VERIFY – проверка выгруженных файлов;
MODE – переключение режима.

В программе имеются 3 режима работы:

Режим 1 – 41 984 байта свободной памяти.
Режим 2 – 44 032 байта свободной памяти.
Режим 3 – 44 288 байтов свободной памяти.

Все команды должны завершаться нажатием клавиш "ENTER". Для выхода в основное меню нужно использовать клавишу "BREAK". При переходе на другой режим происходит очистка памяти, т.е. загруженная информация будет уничтожена.

После выбора одного из вышеуказанных режимов программа выдает запрос. Например, нажатие клавиши "D" дает режим DELETE и появляется запрос DELETE FROM... TO...– удалить записи с ... по ....

В ответ необходимо указать номера тех записей, которые нужны пользователю. Например, если нужно уничтожить записи со второй по пятую, то соответственно надо ввести 2 и 5.

COPY-COPY (PIRATE Ø2) – эта программа находится в эксплуатации довольно давно и представляет собой классический образец копировщика. Она не выполняет компрессирование и имеет сравнительно небольшой объем свободной памяти (42 K), но в ней есть несколько оригинальных режимов, которые в ряде случаев делают ее незаменимой.

Перечень команд. Все команды являются ключевыми словами ПЭВМ "Спектрум" и поэтому не набираются по буквам. Они требуют завершения нажатием клавиши "ENTER".

LOAD – выполнить загрузку очередной записи (блока).

LOAD N TO M – выполнить загрузку записей, начиная с номера N до номера M. При этом ранее существовавшие записи с номером N выше будут уничтожены, т.е.LOAD 1 уничтожит все ранее существовавшие записи и подготовит компьютер к загрузке новых.

Возможные вариации этой команды

LOAD N TO;
LOAD TO M.

SAVE – выполнение выгрузки записей на лёнту. Может иметь форму

SAVE N TO M;
SAVE N TO;
SAVE TO M;
SAVE N TO M STEP K.

Параметр K после оператора STEP показывает величину паузы в секундах, которую компьютер выдержит между отдельными блоками.

VERIFY – проверка выгруженных записей. Имеет все те же формы, что и команда LOAD.

LOAD AT ADDR – загрузка блока, начиная с адреса ADDR Например, LOAD AT 16 384 – загрузка экрана.

LIST ADDR – выдача на экран содержимого памяти, начиная с указанного адреса.

POKE ADDR, N – поместить по указанному адресу байт N. Этой командой часто пользуются совместно с командой LIST для внесения изменений в машинный код программы. В приниципе с подобными задачами лучше справляются специализированные программы для отладки машинного кода (мониторы), но наличие такой функции в копировщике делает его весьма удобным.

COPY 16 384 – команда, которая позволяет копировать блоки длиной 49 K.

Этот режим делает данную программу незаменимой во многих случаях. Дело в том, что если программа имеет один значительный блок длиной порядка 49 K и не поддается компресии, то откопировать его копировщиками типа COPY 86/M и TF COPY не удастся, так как он не поместится в памяти. В режиме COPY 16 384 программа COPY – COPY удаляет саму себя из памяти и оставляет только небольшой блок длиной несколько байтов, предназначенный для выгрузки загруженного блока. Выгрузка выполняется нажатием клавиши "CAPS SHIFT". Возможна только однократная выгрузка, после чего копировщик необходимо загрузить снова. Разумеется, магнитофон должен быть включен на запись до нажатия клавиши "CAPS SHIFT".

В заключение следует отметить, что нами были рассмотрены только три наиболее распространенные копирующие программы. На самом деле их гораздо больше, но принципы их работы отличаются незначительно. В то же время, необходимо отметить, что с помощью копировщиков можно копировать только те программы, в которых не принято специальных мер для защиты от копирования. Многие фирмы такие меры применяют. Это, например, спидлок (ускоренный загрузчик), джеркитон (пилоттон, сопровождающийся рывками), блоки избыточной длины (более 50 K), фальшхэдеры (блоки длиной 17 байтов, которые воспринимаются копировщиком как заголовки очередных блоков в то время как они таковыми не

являются), укороченный пилоттон, удлиненный пилоттон, слишком широкий или слишком узкий пилоттон, блоки с замеряемой паузой между ними и многие другие [15, 22]. Принцип их действия состоит в том, что первый блок программы является Бейсик-загрузчиком, подготавливающим загрузку второго блока. Второй блок (в машинных кодах) подготавливает нестандартную загрузку прочих блоков, которые уже не могут быть загружены в то место, где предварительно не отработал второй блок, в том числе и в копировщик. Копирование таких программ – довольно трудоемкий процесс, но со многими защитами справляется, например, специальная программа LERM-7 (TC-7) [15, 22].

Другой подход к таким программам состоит в том, что используются специальные периферийные устройства. Они предназначены для того, чтобы остановить программу в каком-либо месте, а затем сделать полную выгрузку всей памяти (полный дамп). Также известны устройства типа MULTIFACE, позволяющие, кроме этого, производить выгрузку экрана в любом месте программы, вносить в программу изменения и продолжать исполнение программы с места прерывания. Аналогичными устройствами иногда снабжаются и дисковые системы, например БЕТА-ДИСК ИНТЕРФЕЙС (TR DOC) фирмы "Technology Research". Он имеет специальную кнопку, называемую MAGIC BUTTON (волшебная кнопка), для прерывания работы программы и сброса ее на диск. Однако запустить откопированную таким образом программу через LOAD не удается. Для этого интерфейс реализует нестандартный подход, и запуск выполняется по команде GO TO "имя".

В [15] предлагается способ копирования защищенных программ с помощью двух магнитофонов Один из них включается на вход компьютера, а другой – на выход. Компьютер считывает информацию бит за битом и тут же транслирует ее на выход, при этом он выполняет роль компаратора и выравнивает фронты сигналов. Для этого необходима несложная программа в машинных кодах которая приводится ниже.

```
10 DIM A(12)
20 FOR I=1 TO 12
30 READ A(I)
40 POKE (29999+I),A(I)
50 NEXT I
60 DATA 243,175,219,254,31,31,31,211,254,195,50,117
70 RANDOMIZE USR 30000
```

Отметим, что предложенный метод не очень надежно работает на фирменных "Спектрумах", так как у них выход и вход имеют электрическую связь, но на компьютерах "Таймекс" и на большинстве самодельных дает хороший результат.

2.4.5. Подключение компьютера к телевизору

Фирменные компьютеры работают в дециметровом диапазоне (36-й канал), поэтому подключать их нужно ко входу ДМВ телевизора. При этом надо помнить, что цветовые сигналы идут в системе PAL, поэтому на отечественном телевизоре, работающем в системе СЕКАМ, изображение будет не цветным, а черно-белым (полутоновым). Одним из вариантов выхода из этой ситуации является встраивание в телевизор декодера ПАЛ/СЕКАМ, что пригодится пользователю и при эксплуатации видеомагнитофона.

У большинства самодельных компьютеров, как уже упоминалось выше на телевизор поступают сигналы R, G, B и синхросмесь. Последняя подается на вход субмодуля синхронизации (по цепи "видео"), а сигналы R (красный), G (зеленый), B (синий) – в модуль цветности, На соответствующие входы формирователя первичных цветовых сигналов (обычно это микросхема К174АФ5). Желательно отключить субмодуль радиоканала. Для этого, например в телевизорах 3 УСЦТ, предусмотрена специальная блокировка, которую достаточно замкнуть на корпус (разумеется, это удобнее сделать через тумблер, предварительно утановленный на переднюю или заднюю панель). Добавим также, что для исключения всевозможных неприятностей подключение всех вышеупомянутых сигналов следует производить через "развязывающие" конденсаторы емкостью от 0,22 до 2,2 мкФ.

Для подключения к черно-белому телевизору синхросмесь и сигналы R, G, B необходимо "замешать". Лучше, конечно, для этой цели использовать простой сумматор на основе операционного усилителя, но можно обойтись и обыкновенными резисторами (порядка 300 Ом). Через них сигналы R, G и B соединяются между собой. В точку их соединения напрямую (не через резистор) подводится и синхросмесь. В результате чего получается полный видеосигнал, который можно подавать на вход блока радиоканала (желательно через "развязывающий" конденсатор). При этом выходной каскад блока радиоканала лучше отключить. Суммирующую резистивную матрицу удобно поместить непосредственно в корпус разъема, подключаемого к компьютеру, и тогда для связи с телевизором будет достаточно двух проводов [15, 22].

2.4.6. Использование другого периферийного оборудования

К ПЭВМ "Спектрум" также могут быть подключены накопитель на гибких магнитных дисках (НГМД) и микродрайв ("ZX-Microdrive").

"ZX-Microdrive" – устройство, работающее с высокоскоростной закольцованной магнитной лентой, находящейся на сменных картриджах. Емкость картриджа порядка 100 К. Как уже отмечалось,

оставаясь по сути ленточным носителем, микродрайв создает эффект работы с дисководом. Одновременно к компьютеру можно подключить до восьми таких устройств. В отличие от магнитофона микродрайв не может управляться командами SAVE, VERIFY, LOAD и MERGE, а лишь командами PRINT, LIST, INPUT и INKEY$. Совместно с интерфейсом-1 он может применяться для организации сети из нескольких компьютеров "ZX-Spectrum" [15, 18].

Интерфейс-1 – это устройство, выполняющее одновременно три функции – контроллера локальной сети, порта RS-232 и контроллера микродрайва. Выпускается той же фирмой "Sinclair Research LTD". Имеет собственное ПЗУ и служит для подключения компьютера к другим компьютерам (практически любым), к глобальной сети (через модем) и для объединения класса "Спектрумов" в локальную сеть (до 64 шт.). Команды интерфейса-1 уже не являются операторами Бейсика как языка программирования, а командами операционной системы [15, 18].

Стандартным интерфейсом для "ZX-Spectrum" является RS-232, посредством которого подключаются клавиатура, принтер и любые другие устройства, отвечающие стандартам этого интерфейса. При работе с такими устройствами могут использоваться имеющиеся на клавиатуре дополнительные ключевые слова: OPEN#, CLOSE#, MOVE, ERASE, CAT и FORMAT. Подключение же к ПЭВМ "Спектрум" НГМД требует специального контроллера [23].

2.5. Описание игр компьютера

Для ПЭВМ "Спектрум" разработано большое количество игровых программ. Например, широко используются такие игры, как The weeks in Paradise (Три недели в раю), Robin from the wood (Робин из леса), Psytron, Nonterraqurcus (Нонтеррак), Zorro (Зорро), Caezar the Cat (Кот Цезарь), Tapper (Бармен), The way of the exploding fist (Путь разрушающего кулака), Knighf Zore, Hodu of yesod, Chimera (Химера), Gunfright (Страшный револьвер), Batman, Pyiamasama, Cauldron (Котел), Cauldron 2 (Котел 2), Sir Fred (Сэр Фред), Fairlight, Lightforce, Cobra (Stalone), Twister, Showjumping (Скачки), Basketball, Academy, Uridium, Flying dragon, Straquake, Ocean (Океан), Tamahawk, Skyfox, Ace (Ас), Flyer fox, Spitfire 40, Delta wing (Планер), Gyroscope (Волчок), Potty proffessor, Heavy on the magic (Положись на магию), Rogue trooper (Бродячий солдат), Nasferatu, Fjrelord, Strike force cobra, 2112 A.D., Treasure island (Остров сокровищ), Punchy, Melbourne draw.

Предлагаем описание некоторых из перечисленных выше игр [26].

Robin from the wood (Робин из леса). История Робина Гуда вновь оживает в этой приключенческой игре. Так как легенда тех времен существовала главным образом в устной форме, возникает разница

в вариантах и некоторая произвольность в трактовке фактов. Данная легенда рассказывает о Робине, сыне Алерика, который был стражником Серебряной Стрелы, являвшейся для саксонских племен символом свободы, мира. Но случилось несчастье – на саксонские земли напали норманны, уничтожая все на своем пути. Управляющий графством Ноттингем убил Алерика и завладел Серебряной Стрелой. Для саксонцев наступили тяжкие времена... Когда Робин вырос и смог овладеть оружием, он стал для норманно грозным противником. Он отнимал у захватчиков награбленное добро и раздавал бедным. Воспитанный в лесу, он без труда ускользал от погони. Шериф из Ноттингема, зная о том, что собой представляет Серебряная Стрела для саксонцев, а также о том, что Робин Гуд не сможет уклониться от вызова на поединок, решил обманным путем заманить Робина в город. Шериф объявил состязание по стрельбе из лука, в котором главной наградой победителю должна быть та самая Серебряная Стрела. Именно с этого места легенды и начинается игра.

Перед тем как принять участие в состязании с другими стрелками, играющий должен выполнить несколько других, также трудных и опасных заданий. Чтобы приступить к состязаниям и не быть опознанным, он должен иметь три магические стрелы. Они находятся у старого умного Анта. Кроме того, у Анта находятся лук и меч играющего. Чтобы их получить, игрок должен дать взамен три кошелька с золотом за каждый вид оружия. Однако для того, чтобы добыть кошельки с золотом нужно будет вступить в борьбу с норманнами, составляющими эскорт злого и богатого Епископа из Петерборо.

Территория игрового поля разделена на три части. Наибольшая из них – это лес, потом – замок, в котором будут происходить состязания, и затем – подземелье замка куда могут бросить. Лес – это сложный лабиринт, образуемый зарослями и деревьями по которому бегают красные кабаны, встреча с которыми не сулит ничего хорошего.

Робин начинает игру, будучи вооруженным только коротко палкой однако ее достаточно для того, чтобы победить шныряющих повсюду норманнских воинов которые имеют приказ убить Робина и стреляют без предупреждения. В лесу можно также встретить собирающую зелье Колдунью, которая будет требовать дани, а если игрок не выполнит ее желание, – она бросит его в подземелье выбраться из которого он сможет только в том случае, если найдет ключ до того, как окажется в подземелье. Однако если он к моменту встречи с Колдуньей уже соберет достаточное количество зелья (оно выглядит, как цветочки), то в награду Колдунья перенесет его в другую часть леса. К сожалению, она также заберет и с таким трудом добытые кошельки с золотом поэтому встречи с ней нужно избегать, несмотря на то, что выглядит она довольно симпатично.

Зелье, колчан со стрелами, а также премию в виде дополнительной жизни можно найти в лесу. Собранные и уже имеющиеся предметы высвечиваются в нижней части экрана.

Выше находится сложный орнамент (стилизованные оленьи рога), цвет которого символизирует состояние здоровья Робина. Каждое ранение, полученное им в борьбе, уменьшает его силы. Если он не будет достаточно ловким и сильным, то погибнет, не успев выполнить задание. Очень полезно будет навестить Старого Пустынника, который излечивает раны, однако нужно помнить о том, что Пустынник не переносит вблизи от своего дома никакого оружия, и в этом случае встреча с ним может быть очень неприятной. Кроме того, по лесу путешествует Епископ из Петерборо, охраняемый своими норманнскими стражниками. Достаточно лишить его эскорта, и Епископ убегает, бросая полные кошельки золота.

Иногда во время своих путешествий Робин может встретиться с самим ноттингемским Шерифом, что всегда кончается заточением в подземелье [26].

Psytron (Псайтрон). Сам по себе сценарий игры заключается в охране космического поселения от неприятельских атак. Чужаки будут стараться бомбардировать здания, оборудование и провести своих разрушающих друидов в туннель.

Однако есть то, что отличает игру Psytron, разработанную фирмой "Бейлонд", от множества подобных игр: не встречавшаяся ранее сложность и высокий класс графики. Картину битвы можем видеть на 10 экранах, представляющих различные области космической колонии. Каждая из этих кратин может быть вызвана на монитор.

Первый уровень может показаться легким: все, что нужно сделать, – это поймать друидов в туннеле. Если пять раз подряд результаты борьбы играющего достигнут 50%, игра перейдет в следующий уровень, т.е. ему нужно будет расстреливать вражеские космические корабли. По мере продвижения вперед будет появляться большое количество элементов, сложностей, которые должны быть преодолены по дороге к последнему уровню. Необходимо бороться не только с захватчиками, но также помнить о том, что нужно уменьшить число разрушаемых ими зданий, и чтобы не гибли люди. Если, например, будет разрушена плотина, можно потерять большое количество необходимых товаров [26].

Tapper (Бармен). Если кто-то не согласен, что барменом быть трудно, может попробовать сам, став участником этой игры.

Играющему нужно обслуживать гостей в чрезвычайно популярном пивном баре. Самое печальное то, что клиенты – представители так называемого "темного мира": ковбои, панки, люди, вполне приличные на первый взгляд, однако достаточно агрессивные. Все они хотят только одного – пепси !

Если кто-нибудь из них дойдет до конца строки и не получит

своего стаканчика пепси их терпение моментально иссякнет, и несчастный бармен сам поедет по гладкой поверхности стойки. Работодатель бармена не одобряет таких инцидентов, так же как и того, что бьется посуда. Нужно успевать подхватывать пустые кружки и помнить о том, чтобы не давать большего количества напитков, чем требуется. Пять упущений влекут за собой увольнение с работы – конец игры. Нужно работать исключительно внимательно. Но шеф может, в порядке исключения, забыть о каком-либо упущении.

Это еще не все. Изредка в бар влетает исключительно несимпатичный парень с маской на физиономии Он всыпает яд во все кружки с пепси, за исключением одной. Игрок должен указать на ту кружку, которую бандит не держал в руках [26]

Gunfright (Страшный револьвер). В игре довольно реалистично изображены аллеи и домики обычного западноамериканского городка. Называется этот городок Черный Камень. По улицам рассыпаны груды камней и виден интерьер зданий. Чтобы посмотреть на экран под разными ракурсами, нужно нажать клавишу "Z" или "CAPS".

Играющий – шериф по прозвищу Быстросхват, два острых глаза блестят из-под шляпы Цель его – патрулировать город, где шатаются бродяги и вооруженные бандиты. Шериф вооружен шестизарядным револьвером.

Игра начинается со сцены падения мешков с деньгами; их надо набрать как можно больше, потому что придется платить за патроны, лошадей, а также платить штрафы.

Следующая сцена начинается с того, что игрок может перемещать шерифа, стоящего в дверях тюрьмы, во всех направлениях и стрелять. Слева на экране показано объявление – "разыскивается опасный преступник". Количество пулек под ним – это число оставшихся жизней. Еще ниже – сумма заработанных денег. В самой нижней части экрана показывается барабан с оставшимися пулями и телеграфный бланк с ценами на пули и лошадей, а также уровней штрафов.

Задача играющего – быстро проходить по городу, наполненному опасностями. Но самое опасное – это не бандиты, а мирные богобоязненные обитатели городка которые бесцельно шляются по улицам, столкновение с которыми лишает шерифа жизни и 150 долларов.

Вскоре играющий обнаружит вооруженного налетчика, который выглядит довольно устрашающе. Однако если игрок выстрелит и попадет в него, то с удивлением обнаружит, что лишился еще 150 долларов. Этот парень всего лишь указывал своим револьвером направление, где находится ближайший бандит.

С помощью других обитателей городка игрок в конце концов доберется до Билла Быка, за поимку которого выплачивается преми

в размере 350 долларов. Не надо ждать, пока он начнет стрелять, игрок должен постараться выстрелить в него сам – и игра перейдет на следующий уровень. Играющему предстоит столкнуться с ним лицом к лицу, и теперь результат зависит только от реакции и меткости игрока. Если ему удастся поразить бандита первым, он возвращается к начальной картине, но теперь на очереди Билли Козленок, за которого платят уже больше – 700 долларов.

Еще одно: играющий может найти лошадь, которую зовут Панто, и сидя на ней, можно спокойно проскакать полгородка, сметая все на своем пути, но, к сожалению, лошадь – удовольствие дорогое. Однако если играющему посчастливится встретить бандитов верхом на коне, он может пуститься за ними в погоню [26].

Sir Fred (Сэр Фред). Надев доспехи, можно смело пускаться по скрытой туманом древности стране сварливых королей, прекрасных дам и рыцарей без страха и упрека.

Сэр Фред попал в эту страну в то время, когда все остальные благородные рыцари находились в дальних странствиях, а оставшиеся забросили свое ремесло и спокойно доживали жизнь в своих замках, построенных на золото драконов. Когда злой сэр Гуго Дьюни похитил принцессу, король, ее отец, встал перед сложной проблемой, как найти благородного рыцаря, который бы проник в замок негодяя и спас дочку. Королю пришлось остановить свой выбор на сэре Фреде. Хотя он и не самый лучший рыцарь королевства, но старательный.

Трудности начинаются с самых первых секунд игры. На первом экране сэр Фред, запакованный в доспехи, стоит на краю пропасти, по дну которой ползет змея. Ее укусы лишают неосторожных рыцарей сил. Можно перепрыгнуть через пропасть, раскачавшись на одной из веревок, а можно просто броситься вниз, перепрыгнуть через змею и выбраться наверх по виноградной лозе, которая свешивается с края пропасти вниз. Кстати, если есть желание выиграть в этой игре, надо быстренько научиться раскачиваться на веревках. На большей части экранов имеется по крайней мере одна веревка. Залезть на такой канат очень просто, а вот раскачаться на нем, вовремя перенося тяжесть тела с одной стороны на другую, – это уже искусство, на овладение которым может уйти много времени. К несчастью, пока играющий им овладеет, большинство полезных предметов ему уже недоступно.

Уйдя с первого экрана направо, играющий окажется перед рвом с водой. Многие предметы находятся на дне водоемов, так что Фреду надо еще уметь плавать. Если играющий не столкнулся с эффектом инерции на первом экране, то он ощутит его, как только "спрыгнет" в ров. В воде на него действует выталкивающая сила. Чтобы "нырнуть", он должен нажать кнопку "вперед" и одновременно удержать кнопку "вниз". Управляя таким образом пловцом, можно

убедиться, что проникнуть в подводную пещеру довольно трудно.

Эффект инерции чувствуется и на суше. Во время движения надо недолго держать кнопку "вперед", иначе Фред рискует врезаться на полном ходу в какое-нибудь препятствие. Но вернемся ко рву. Сцена надо рвом имеет два плана. Сперва играющий должен по стенке замка подняться наверх и, раскачавшись там на веревке, запрыгнуть на облако, чтобы включить там рубильник. При этом необходимо уклоняться от ударов молнии. Любая неточность – и игрок летит в ров.

В замок можно попасть и через подводный туннель, захватив по дороге горсть камней, если повезет. Камни могут пригодиться. Проплыв через туннель, играющий окажется на экране с двумя лагунами. Нужно выбраться из первой и нырнуть во вторую.

Осторожно! Вполне реальная опасность наткнуться на гигантского осьминога! Если повезет, на дне этой лагуны играющий найдет меч Фреда. Но уверенности в этом никакой, так как он может находиться и в каком-нибудь из 58 мест, где лежат разные предметы. Меч – главное оружие Фреда в игре; как и все остальные предметы, его можно взять с помощью клавиши "SELECT", а в действие он приводится после того, как к обозначающему меч символу на нижней половине экрана подводят курсор и нажимают клавишу "USE". Все остальные предметы, которые несет Фред, тоже показаны в нижней части экрана в "окошках". С мечом в руках можно нападать и защищаться.

Первый, сделавший выпад, имеет преимущество. У врагов может быть 7 уровней фехтовального мастерства, и справиться с ними даже на самом первом уровне, не так просто! Чтобы победить противника, его надо поразить три раза, если только не удалось припереть его к стенке. Другое важное оружие – лук и стрелы. (О том, как добыть его, описывается ниже). После попадания в замок события убыстряются, и испытания ужесточаются.

Чтобы попасть в замок, возможны три варианта.

Вариант 1. Взять веревку со дна пропасти. Выбраться из пропасти и идти налево. Там сбросить веревку с обрыва вниз и спуститься по ней на дно. Взять лук и стрелы. Раскачавшись на веревке, перепрыгнуть через желтую скалу слева. Перейдя на левый экран, подняться по цепи подвесного моста и, пригнувшись, прыгнуть в замок.

Вариант 2. Взять мясо со дна пропасти. Бросить его в ров, чтобы отвлечь пиранью. Проплыть через туннель под стеной. Из лагуны с осьминогом взять спички и снова выбраться в ров. Забраться по виноградной лозе на стену, раскачавшись на веревке, запрыгнуть на облако, включить рубильник и пройти в открывшийся проход. Подойти к трезубцу, на котором лежит бомба. Поджечь бомбу спичками. Бомба взорвется и уничтожит препятствие. Перепрыгнув через обломки, войти в замок.

Вариант 3. Взять меч в лагуне с осьминогом. Подняться по стене замка, залезть на облако, включить рубильник и, пройдя в открывшийся проход, взять бутылку. Снова поднырнуть под стену На экране справа от осьминога подозвать паромщика с помощью бутылки. Дальше идти направо. Убрать фехтовальщика и пройти в замок, поднявшись по свисающей сверху веревке [26].

Fairlight. Это первая игра в трилогии о земле Фарлайта, и игрок будет действовать в окрестностях замка.

На безбрежной равнине ничем не примечательного плато, по ту сторону непроходимого густого леса расположен заколдованный замок. В замке томится волшебник, игрок должен найти волшебную книгу, чтобы вызволить его из темницы.

Персонаж играющего – это путешественник и исследователь, одетый в плащ и вооруженный мечом, чтобы отбиться от антисоциальных типов. Фигурка может двигаться в четырех направлениях, прыгать и поднимать различные предметы, вступать в бой.

Играющий может передвигать предметы, толкая их, ставя стулья на столы, чтобы добраться до высоко расположенных дверей или окон а также переставлять мебель. Проблемы, которые надо решить, обычно просты. По виду ключей ясно, к какой двери они подходят. После того, как откроется очередная потайная дверь, можно проникнуть в глубины замка. Охранников и троллей, которые, противодействуя, размахивают большими дубинками, можно перехитрить или вступить с ними в бой. Реализм схваток включает сопоставление сил сражающихся.

Каждый предмет и персонаж в игре имеют вес. Предметы, которые несет игрок, изображаются на маленьком свитке в углу экрана по одному за раз. Можно нести до пяти предметов одновременно, но их вес играет существенную роль. Возможно, силы игрока хватит только на меньшее количество Аналогично удар дубинки тролля намного чувствительнее, чем наскоки маленького охранника [26].

Lightforce. Играющий стреляет по предметам, увертываясь от них, при этом нужна быстрая реакция. Если играющий выдержит несколько начальных уровней, то это очень неплохо!

Имея на вооружении несколько межпланетных крейсеров, нужно уничтожить целый флот инопланетных кораблей, а также их базы, заводы и прочие строения. Некоторые корабли пускают торпеды, что очень неприятно. В игре много различных конструкций. Играющий начинает свой путь через джунгли одной из планет, потом ему предстоит пройти через водный мир, индустриальную область и через реку. Наивысший счет достигается не просто глубиной проникновения на территорию противника, но также и различного рода призами. Обычно премия назначается при разрушении всех зданий комплекса. Игра очень проста на вид, однако "выжить" в ней далеко

не так просто, как кажется. Вражеские эскадрильи летят в сложном боевом порядке. Они быстро перестраиваются, применяют тактические уловки, которые для каждого типа кораблей свои.

Полезные советы. Сначала о поведении врагов: Зеленые летающие тарелки двигаются зигзагами вниз, подбить их нельзя.

Красные летающие тарелки: кружат вокруг игрока, некоторые пускают ракеты, остальные бросают мины, все это в его направлении.

Голубые летающие "додасы": следуют постоянным курсом их легко подбить.

Желтые "плоп-плопы": они следуют тем же курсом, что и голубые "додасы", но внезапно поворачивают и летят прямо на игрока.

Зеленые астероиды: их нельзя подстрелить. Они появляются всякий раз, когда есть возможность получения премии.

Белые астероиды: следят за игроком, их нужно уничтожать пораньше.

Зероиды: появляются из-за спины игрока и летят прямо на него.

Истребители: некоторые появляются сбоку, а некоторые стреляют из-за рамки экрана.

Теперь о последовательности событий: ПОЯС АСТЕРОИДОВ – зеленые астероиды, комплекс зданий, желтые астероиды, комплекс зданий, три ряда голубых летающих тарелок, желтый астероид, здание, белый астероид, красные летающие тарелки (две волны), здание, желтый астероид и, наконец, балый астероид. ПЛАНЕТА ДЖУНГЛЕЙ – зероиды, зеленые тарелки, здания, истребители, снова истребители, опять истребители, взлетная полоса, здание, кресты, призовая жизнь, зеленые тарелки, истребители, здания, кресты, истребители, призовая жизнь, зеленые тарелки, зероиды и зеленые астероиды. ОРБИТАЛЬНАЯ ПЛАТФОРМА: Зеленые астероиды, красные тарелки, желтые астероиды, белые астероиды, призовая жизнь, зероиды, 4 волны красных тарелок, зеленые астероиды, красные тарелки, желтые астероиды, зеленые астероиды, красные тарелки, зероиды, красные тарелки, красные тарелки, стреляющие ракетами, снова тарелки, желтые астероиды, красные, красные зероиды, зеленые зероиды, призовая жизнь. ЛЕДЯНАЯ ЖИЗНЬ: хотя некоторые типы могут слегка отличаться, все нападающие ведут себя точно так же, как и над ПЛАНЕТОЙ ДЖУНГЛЕЙ [26].

Basketball (Баскетбол). Можно играть с партнером или с компьютером, руководя командой из 4 игроков. Действия очень быстры, и играющему придется основательно потренироваться, если он хочет противостоять компьютеру, даже на самом первом уровне. К счастью, есть режим тренировки.

Чтобы добиться успехов в игре, недостаточно просто быстро бегать и бросать мяч в корзину, необходимо научиться передавать мяч партнеру. Играющий может управлять игроком, который в

настоящий момент ближе всего к мячу или владеет им. Такой игрок отличается повышенной яростью. Нажав кнопку "огонь", играющий заставит игрока подпрыгнуть; продолжая ее удерживать, можно заставить подпрыгивать всех игроков по очереди. Если во время прыжка нужного играющему члену команды он отпустит кнопку, то контроль перейдет к нему. Пока играющий управляет этим игроком, машина берет на себя заботу об остальных. Обычно они передвигаются довольно рационально. Хотя иногда их движения поражают бестолковостью. Игра быстрая, довольно сложная [26].

Tamahawk (Апаш). В программе есть возможность тренировки. В вертолет играющего никто не стреляет, хотя вражеские вертолеты, как камикадзе, все время пытаются столкнуться с ним. Есть и другие варианты. Полностью игра развертывается на всей карте: некоторые ее квадраты контролируются играющим, а другие – противником. Необходимо выбить неприятеля из его позиций, не дав ему при этом проделать то же самое с играющим. Так как каждый квадрат содержит 8 целей, то процесс может длиться очень долго. Авторы считают, что их рекорд – это 4 очищенных за 22 минуты квадрата, и что для того, чтобы очистить все остальное пространство, потребуется не менее 20 часов.

Пусть играющий попробует "летать" ночью или в тумане (а это значит, что он не видит предмета, пока с ним не столкнется). Он убедится, что самое трудное – это уменьшить скорость. Лучше всего поднять нос вертолета вверх, но тогда он теряет и высоту. Если же вертолет летит слишком высоко, и он не может рассмотреть мишень, то приходится пикировать. При этом скорость нарастает так быстро, что играющий не успевает опомниться, как у его вертолета отрываются лопасти несущего винта. С помощью "газа" играющий поддерживает высоту а с помощью джойстика – разгоняет или тормозит.

Что касается турбулентности и воздушных потоков, то вертолет намного более устойчив к ним. С ними не надо бороться, их надо "оседлать". Кроме этого, на малой высоте турбулентность практически отсутствует, так что от нее можно избавиться, спустившись вниз Правда, внизу находятся линии электропередачи...

Настоящий вертолет "Апаш" стабилизируется с помощью компьютерной системы, без нее вообще невозможно летать, и он очень мощно вооружен: 16 ракет "Хеллфайер", 30-миллиметровое орудие и 19 навесных пусковых установок для реактивных снарядов калибра 70 мм. Создателям программы это показалось слишком много, и они ограничились орудием, двумя ракетами и одной пусковой установкой для снарядов. Впрочем, играющий всегда может "приземлиться" и возобновить свои запасы.

У настоящего вертолета имеется лазерный прицел в носовой части Этим прицелом можно управлять с помощью системы,

носящей название Интегральной Системы Связи Шлема Пилота с Прицелом. Это означает: пилот двигает головой и одновременно перемещает прицел. Тактика боевых действий такова. Пилот на мгновение поднимает из-за холма свой вертолет, и его компьютер запоминает ландшафт в виде картинки. Затем он спускается вниз, не торопясь, наводит прицел по изображению и снова взмывает вверх лишь на одно мгновение, чтобы выпустить ракету. После этого снова можно прятаться, ракета сама найдет цель по лазерному лучу. Ночью все это делается с помощью приборов ночного видения.

Если не считать шлемонаведения, оружие в игре работает почти так же. Нужно отметить: цели на радаре появляются задолго до появления самой цели на главном экране. Как только цель из линейчатой превратится в сплошную, можно выпустить ракету. Она попадает, даже если цель будет за горами, деревьями или облаками. Вражеские вертолеты не являются "Апашами". Однако пилоты их отчаянно смелы, и новый вертолет появляется сразу после того, как сбит его предшественник. То, что они все-таки появляются по одному, лишь отчасти облегчает игру. Они всегда летят на высоте вертолета играющего, а радиус действия их ракет 4000 футов. Так как ракеты играющего настигают цели за 3 мили, то это – игра в одни ворота. Правда, стоит помнить, что вражеские вертолеты появляются снова и снова, а вертолет играющего – в единственном экземпляре [26].

10 советов начинающему пилоту.

1. Для быстрого набора высоты во время боя нужно выжать газ до отказа и опустить нос вертолета на 30 градусов Меньше чем за 6 секунд играющий наберет 100 метров.

2. Не надо бояться посадки на землю. Можно приземлиться, где угодно, а потом добираться до базы, руля по граниту.

3. Для атаки следует выбирать среднюю высоту, так как все равно придется опускать нос вертолета, и если играющий будет на большой высоте, то его вертолет слишком быстро наберет недопустимую скорость.

4. Танки можно подбить только ракетами.

5. Наиболее удобный способ сбрасывать скорость – делать быстрые повороты вправо–влево.

6. Чтобы круче развернуться, нужно поднять нос вертолета вверх.

7. Проще всего поразить вертолет противника, когда он поворачивает. Через горы он пролететь не сможет, поэтому следует устраивать засады за горами и ждать, пока он будет маневрировать

8. Целесообразно сразу набрать ракет и подняться над базой на высоту 512 футов. Поворачиваясь на месте, "засечь" все цели в радиусе 3 мили. Уничтожить их все, спускаясь вниз за новыми ракетами по мере необходимости.

9. В стратегическом варианте игры начальное положение сторон все время разное. Если выбрать игру попроще, следует прервать ее в самом начале, пока позиция не понравится.

10. Если удастся занять целиком весь ряд вражеских позиций, то противник уже не сможет их вернуть. Надо быть осторожным: противник тоже может поступить подобным образом! [26].

Treasure island (Остров Сокровищ). ...Сначала Джим Хоукинс на пути к борту "Эспаньолы" должен избежать смертельных объятий Слепого Пью. Затем ему нужно бояться разоблачения в тот момент, когда он подслушивает разговор мятежной команды корабля. И только после этого начинаются собственно приключения на острове Сокровищ.

Поиск начинается от форта, который искатели приключений построили на острове. Джим должен опередить пиратов и найти пещеру Бена Ганна. Поиски ведутся в соответствии с картой, части которой появляются на экране. Всего таких частей – 64.

Когда найдена пещера, сокровища обнаружены, для завершения игры их нужно перенести на корабль, спрятанный в бухте.

В конце каждой игры приводится счет, отражающий скорость, с которой сокровища были найдены, а также бойцовский дух игрока Чтобы добиться абсолютного результата 100 %, придется изрядно потрудиться.

Управляющие клавиши:

"Q" – вверх;	"O" – влево;
"A" – вниз;	"P" – вправо.

Для получения полного успеха недостаточно избегать ловушек пиратов. Необходимо приблизиться к пирату на такое расстояние, чтобы он метнул в Джима кортик, и при этом успеть увернуться.

Чтобы взять что-то, Джим должен быть помещен на подбираемый объект. Бросая кортик, Джим должен двигаться в направлении броска; в момент броска нажимается клавиша "M" или кнопка "Огонь" джойстика.

При нажатии клавиши "H" в игре наступает пауза, а клавиши "T" – игра продолжается. Одновременное нажатие "D" и "T" вызывает рестарт [26].

Skyfox (Скайфокс). Истребитель "Скайфокс" – это настоящий летающий арсенал, управление которым может доставлять много часов острых ощущений, когда играющий будет отбивать атаку за атакой вражеских сил, направленных против него таинственным кораблем-маткой.

Некоторое представление об общих возможностях этой игры можно получить, просматривая меню, которое предлагает на выбор 7 различного рода тренировочных полетов, необходимых, чтобы приготовиться к настоящей игре. Можно "висеть на хвосте" у вражеского истребителя или сталкиваться со всевозможными ситуациями, которыми насыщена "полноразмерная" игра.

На приборной доске находятся не только лазер непрерывного действия, но и управляемые ракеты, ракеты с тепловым наведением и прочие смертоносные орудия.

В рубке также есть бортовой компьютер, который по запросу играющего в любой момент выдает информацию о его местоположении, об общем счете игры, а также о расположении вражеских сил.

Очень полезен автопилот, который выводит к ближайшему месту скопления противника.

Различной информацией снабжают радары, координатные мониторы, указатели горючего, скорости, высоты, состояния защитного поля. Во время игры необходимо уничтожить 8 атак противника, причем с каждым разом это становится все труднее сделать. Задача осложняется еще и тем, что необходимо постоянно охранять от уничтожения собственную базу, при разрушении которой отказывает компьютер. При этом играющий лишается ценной информации о расположении противника и его силах

Даже при самом "маленьком" вторжении силы противника во много раз превосходят силы играющего. Но он в состоянии нанести врагу немалый урон перед тем, как бордюр вспыхнет и замигает красным цветом, сигнализируя о том, что защитное поле уничтожено и увлекательный полет закончен.

Лазерная пушка несет неограниченный боекомплект, а вот с управляемыми и тепловыми ракетами надо обращаться экономично! Хуже всего, если, направив теплонаводящуюся ракету на скопления противника, играющий неосторожно пересечет ее путь Вид собственной ракеты, возвращающейся обратно подобно бумерангу, не доставит играющему никакого удовольствия и нисколько не облегчит и без того трудного положения.

Для того чтобы научиться проделывать ощутимые бреши в рядах нападающих на более высоких уровнях, требуется известная усидчивость и увлеченность. Если удастся даже просто бросить взгляд на корабль-матку, то уже можно поздравить себя с немалыми успехами. Если же играющий наберет для игры вариант 4 "массового" вторжения, то, скорее всего, единственное, что ему покажет экран, – это мигающий красный бордюр. Эта картинка быстро становится очень знакомой.

Сама по себе графика игры достаточно приемлема, хотя искажения перспективы при приближении к атакуемой цели, часто сбивают с толка. Очень часто при приближении к цели на прямом курсе они вдруг резко смещаются вправо или влево. Причиной этого могут быть уставшие от игры пальцы, подрагивающие на ручках управления.

Множество уровней сложности позволяет поддерживать интерес к игре и после того, как притупится новизна впечатлений. "Скайфокс" – одна из самых лучших насыщенных военными действиями игр с истребителем в главной роли для ПЭВМ "Спектрум".

ПРОГРАММАТОР ДЛЯ ПРОГРАММИРОВАНИЯ ПЗУ

Программирование репрограммируемых (перепрограммируемых) ПЗУ (РПЗУ), используемых в ПЭВМ "Спектрум" (а также и в других компьютерах), можно производить с помощью программатора, который описывается ниже. Программатор выполнен в виде приставки к компьютеру "Спектрум", предназначенной для программирования наиболее широко применяемых РПЗУ емкостью от 2 до 32 килобайт отечественного и зарубежного производства (К573РФ2, К573РФ5, К573РФ6, К573РФ8, 2716, 2764, 27128, 27256) [23].

Цоколевка микросхем РПЗУ приведена в табл. П1. Сигналы на выводах микросхем в различных режимах работы даны в табл. П2 (здесь V_{pp} – напряжение программирования, которое указывается на корпусе микросхемы). Схема программатора дана на рис. П1.

Программатор выполнен на программируемом интерфейсном адаптере (ПИА) КР580ВВ55А. Три 8-разрядных потра ПИА с адресами 1F, 3F, 5F (HEX) используются для ввода/вывода данных и формирования адреса ПЗУ. Сигналы управления ПЗУ (CS, OE, PGM) формируются с помощью дополнительного регистра (микросхема 155ИР1) с адресом EF (HEX). Адрес регистра управления ПИА-7F (HEX). Программируемая микросхема ПЗУ вставляется в 28-кантактный разъем (панель), обеспечивающий надежную связь с шинами адреса, данных и сигналами управления ПЗУ. С помощью переключателя П2 осуществляется коммутация адресов и сигналов управления для различных типов ПЗУ. Микросхемы К573РФ2, К573РФ5 и 2716 имеют 24 вывода и вставляются в разъем со смещением (табл. П1). Вставлять и вынимать микросхему из разъема можно только при среднем положении переключателя П1. При таком положении переключателя П1 на него не подаются питающие напряжения. Два других положения переключателя П1 служат для выбора режима "Чтение" и "Запись".

Программа обслуживания программатора состоит из двух частей: первая написана на языке Бейсик, а вторая часть – в машинных кодах (программа PROG+). Программа, написанная на языке Бейсик, поддерживает интерактивный режим обмена с помощью нескольких меню. В блоке машинных кодов содержатся подпрограммы: чтение, запись и тестирование.

Программа обслуживания программатора заносится в память компьютера с магнитной ленты. После загрузки программы на экране телевизора появляется надпись: "Load MONITOR (Y/N)?". Нужно ответить "Y", если используется программа МОНИТОР. После ввода "Y" и "Enter" компьютер будет готов к загрузке МОНИТОРа. Если ввести "N" и "Enter", то можно сразу перейти к началу работы программы. На экране появится выражение:

Таблица П1

32 K × 8	16 K × 8	8 K × 8	2 K × 8			2 K × 8	8 K × 8	16 K × 8	32 K × 8
V_{pp}	V_{pp}	V_{pp}		1	28		V_{cc}	V_{cc}	V_{cc}
A12	A12	A12		2	27		PGM	PGM	A14
A7	A7	A7	A7	3 (1)	(24) 26	V_{cc}	–	A13	A13
A6	A6	A6	A6	4	25	A8	A8	A8	A8
A5	A5	A5	A5	5	24	A9	A9	A9	A9
A4	A4	A4	A4	6	23	V_{pp}	A11	A11	A11
A3	A3	A3	A3	7	22	OE	OE	OE	OE
A2	A2	A2	A2	8	21	A10	A10	A10	A10
A1	A1	A1	A1	9	20	CS	CS	CS	CS
A0	A0	A0	A0	10	19	D7	D7	D7	D7
DO	DO	DO	DO	11	18	D6	D6	D6	D6
D1	D1	D1	D1	12	17	D5	D5	D5	D5
D2	D2	D2	D2	13	16	D4	D4	D4	D4
Gnd	Gnd	Gnd	Gnd	14 (12)	(13) 15	D3	D3	D3	D3

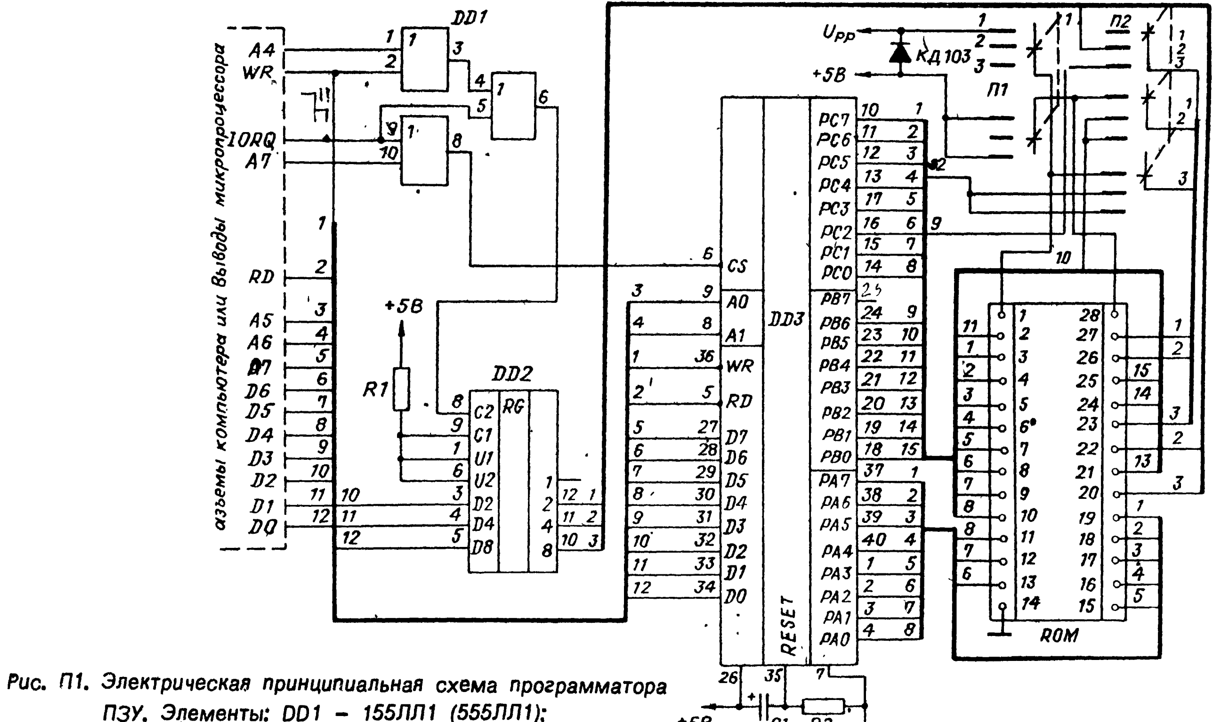


Рис. П1. Электрическая принципиальная схема программатора ПЗУ. Элементы: DD1 – 155ЛЛ1 (555ЛЛ1); DD2 – 155ИР1; DD3 – КР580ВВ55А; C1 – 10,0 мкФ; R1 – 1,0 кОм; R2 – 120 кОм. Положения переключателя П1: 1 – запись; 2 – выключено; 3 – чтение. Положения переключателя П2: 1 – 2716, 573РФ2, 573РФ5; 2 – 2764, 27128, 573РФ4, 573РФ6; 3 – 27256, 573РФ8

"Programming 2716, 2764, 27128, 27256" и меню – "Read or Write or Test or Verify". После этого необходимо включить режим CAPS LOCK и ввести символ выбранного режима:

– R – режим чтения из ПЗУ в память компьютера,

– W – режим программирования ПЗУ информацией, поступающей из памяти компьютера;

– T – режим проверки "чистоты" ПЗУ ("чистая" ПЗУ содержит по всем адресам байт FF (HEX);

– V – режим сравнения содержимого ПЗУ с содержанием памяти компьютера.

Таблица П2

2Kx8		8Kx8		16 Kx8		32Kx8		Номер вывода
Чтение	Запись	Чтение	Запись	Чтение	Запись	Чтение	Запись	
–	–	+5 В	V_{pp}	+5 В	V_{pp}	+5 В	V_{pp}	1
0	_⊓_	0	0	0	0	0	0	20(18)
0	1	0	1	0	1	0	_⊓_	22(20)
+5 В	V_{pp}	A11	A11	A11	A11	A11	A11	23(21)
+5 В	+5 В	–	–	A13	A13	A13	A13	26(24)
–	–	1	‾⊔‾	1	‾⊔‾	A14	A14	27

Объем памяти ПЭВМ, отведенный для всех режимов работы программатора, занимает 32 К (с адреса 8ØØØ (HEX) по FFFF (HEX)).

После выбора режима, например "W" и "Enter", необходимо выбрать тип микросхемы путем установления одного из следующих положений переключателя П2:

– "1" – для программирования микросхем К573РФ2, К573РФ5, 2716;

– "2" – для программирования микросхем К573РФ4, 2764;

– "3" – для программирования микросхем К573РФ6, 27128;

– "4" – для программирования микросхем К573РФ8, 27256.

Затем следует установить начальный и конечный адреса программируемой микросхемы. Адрес вводится в шестнадцатиричном (HEX) виде 4 символами (т.е. два байта), например 01FF. Аналогично вводится начальный адрес рабочей области компьютера (RAM address...), в которой расположена информация для программирования. На вопрос "Correct (Y/N)?" необходимо ответить: "N", "Enter", если набранная информация не требует корректировки, в противном случае следует ответить "Y". Затем на экране появится надпись "Insert chip – Ready? – Press any key".

После этого необходимо вставить в разъем микросхему ПЗУ, установить переключатель П1 в положение "PGM" и нажать любую кнопку для пуска процесса программирования, который графически отображается на экране телевизора. В случае ошибки при программировании на экране появляются ошибочные данные и адрес. Другие режимы работы программатора выполняются аналогично.

```
5 GO TO 30
10 CLEAR 31999: LOAD ""CODE: INPUT"    LOAD MONITOR (Y/N) ?
";M$: IF M$="N" THEN GO TO 30
20 IF M$="Y" THEN LOAD""CODE
30 CLS: PRINT AT 0,10;"PROGRAMMING
50 PRINT AT 2,5;"2716,2764,27128,27256"
60 INPUT "Read or Write or Test or Verify";Q$
70 IF Q$="R" THEN PRINT AT 5,8;"  READ": GO TO 120
80 IF Q$="W" THEN PRINT AT 5,8;" WRITE": GO TO 120
90 IF Q$="T" THEN PRINT AT 5,8;"  TEST": GO TO 120
100 IF Q$="V" THEN PRINT AT 5,8;"VERIFY": GO TO 120
110 GO TO 60
120 INPUT"                INPUT     1-2716, 2-2764, 3-27128,
4-27256",Q
130 IF Q=1 THEN PRINT AT 5,17;"2716": GO SUB 3500: GO
 TO 180
140 IF Q=2 THEN PRINT AT 5,17;"2764": GO SUB 3600: GO
 TO 180
150 IF Q=3 THEN PRINT AT 5,17;"27128": GO SUB 3600: GO
 TO 180
160 IF Q=4 THEN PRINT AT 5,17;"27256": GO SUB 3700: GO
 TO 180
170 GO TO 120
180 PRINT AT 8,1;"ROM adress from ";: DIM W$(4): INPUT
 W$: GO SUB 5000: GO SUB 6000
190 POKE 32117,R: POKE 32190,R: POKE 32314,R
200 POKE 32118,E: POKE 32191,E: POKE 32315,E
250 GO SUB 7000: PRINT I$
260 PRINT AT 8,22;"to ";:DIM W$(4): INPUT W$: GO SUB
5000: GO SUB  6000
270 POKE 32120,R: POKE 32193,R: POKE 32317,R
280 POKE 32121,E: POKE 32194,E: POKE 32318,E
330 GO SUB 7000: PRINT I$
335 IF Q$="T" THEN GO TO 420
340 PRINT AT 10,1;"RAM adress from";: DIM W$(4): INPUT
W$: GO SUB  5000: GO SUB 6000
350 POKE 32123,R: POKE 32196,R: POKE 32334,R
360 POKE 32124,E: POKE 32197,E: POKE 32335,E
410 GO SUB 7000: PRINT I$
420 INPUT "           CORRECT(Y/N)          ";P$
430 IF P$="N" THEN GO TO 30
```

```
440 IF P$="Y" THEN PRINT AT 14,10;"Insert chip
  READY?        Press any key": PAUSE 0: GO TO 460
450 GO TO 420
460 IF Q$="R" THEN GO TO 1000
470 IF Q$="W" THEN GO TO 1500
480 IF Q$="T" THEN GO TO 2000
490 IF Q$="V" THEN GO TO 2500
1000 POKE 32009,119: RANDOMIZE USR 32112
1020 PRINT FLASH 1;AT 21,15;"END": PRINT " For continue
     press any    key : PAUSE 0: GO TO 30
1500 RANDOMIZE USR 32304
1510 IF PEEK 32463=255 THEN PRINT FLASH 1;AT 21,15;"END":
     PRINT " For continue press any key": PAUSE 0: GO TO 30
1520 PRINT AT 18,1;"ROM adress        ": LET U=PEEK 32394+
     256*PEEK 32 395: GO SUB 7010: PRINT AT 18,13;I$
1530 PRINT AT 18,20;"ROM data    ": LET D=PEEK 32095:
     GO SUB 7500: P RINT AT 18,30;S$
1540 PRINT AT 19,1;"RAM adress        ": LET U=PEEK 32400+
     256*PEEK 32 401: GO SUB 7010: PRINT AT 19,13;I$ .
1550 PRINT AT 19,20;"RAM data    ": LET D=PEEK (PEEK
     32400+256*PEEK 32401): GO SUB 7500: PRINT AT 19,30;S$
1560 INPUT "            CONTINUE (Y/N) ?          ";G$
1570 IF G$="N" THEN GO TO 30
1580 IF G$="Y" THEN RANDOMIZE USR 32384: GO TO 1510
2000 POKE 32009,0: POKE 32210,254: POKE 32211,255
2030 RANDOMIZE USR 32176
2040 IF PEEK 32303=255 THEN PRINT FLASH 1;AT 21,15;"END":
     PRINT " For continue press any key : PAUSE 0: GO TO 3
2050 PRINT AT 18,1;"ROM adress        ": LET U=PEEK 32249+
     256*PEEK 32 250: GO SUB 7010: PRINT AT 18,13;I$
2060 PRINT AT 18,20;"ROM data    ": LET D=PEEK 32015:
     GO SUB 7500: P RINT AT 18,30;S$
2070 INPUT "            Continue (Y/N) ?       ";H$
2080 IF H$="Y" THEN RANDOMIZE USR 32240: GO TO 2040
2090 IF H$="N" THEN GO TO 30
2100 GO TO 2070
2500 POKE 32009,0: POKE 32210,0: POKE 32211,190
2530 RANDOMIZE USR 32176
2540 IF PEEK 32303=255 THEN PRINT FLASH 1;AT 21,15;"END":
     PRINT " For continue press any key": PAUSE 0: GO TO 30
2550 PRINT AT 18,1;"ROM adress ": LET U=PEEK 32249+256*
     PEEK32  250: GO SUB 7010: PRINT AT 18,13;I$
2560 PRINT AT 18,20;ROM data     ": LET D=PEEK 32015:
     GO SUB 7500: P RINT AT 18,30;S$
2570 PRINT AT 19,1;"RAM adress        ": LET U=PEEK 32255
     +256*PEEK 32 256: GO SUB 7010: PRINT AT 19,13;I$
2580 PRINT AT 19,20;"RAM data    ": LET D=PEEK (PEEK
     32255+256*PEEK 32256): GO SUB 7500: PRINT AT 19,30;S$
2590 INPUT "            Continue (Y/N) ?       ";J$
2600 IF J$="N" THEN GO TO 30
2610 J$="Y" THEN RANDOMIZE USR 32240: GO TO 2540
2620 GO TO 2590
3500 POKE 32001,8: POKE 32011,10: POKE 32046,2: POKE 32050
```

```
3510 POKE 32068,10: POKE 32082,8: POKE 32091,10: RETURN
3600 POKE 32001,12: POKE 32011,15: POKE 32046,6: POKE 32050,2
3610 POKE 32068,15: POKE 32082,12: POKE 32091,15: RETURN
3700 POKE 32001,8: POKE 32011,9: POKE 32046,0: POKE 32050,2
3710 POKE 32068,9: POKE 32082, 8: POKE 32091,9: RETURN
5010 LET T=0
5020 FOR Y=1 TO 4
5030 LET T=T+16^(4-Y)*(CODE W$(Y)-48-7*(CODE W$(Y)>57))
5040 NEXT Y
5050 POKE 32030,INT (T-(INT (T/256))*256)
5060 POKE 32031,INT (T/256)
5070 RETURN
6000 LET W=PEEK 32030+256*PEEK 32031
6010 LET E=INT (W/256)
6020 LET R=NT (W-(E*256))
6030 RETURN
7000 LET U=PEEK 32030+256*PEEK 32031
7010 DIM I(4): DIM I$(4)
7020 LET I(1)=INT (U/4096)
7030 LET U=U-I(1)*4096
7040 LET I(2)=INT (U/256)
7050 LET U=U-I(2)*256
7060 LET I(3)=INT (U/16)
7070 LET I(4)=U-I(3)*16
7080 FOR O=1 TO 4
7090 LET I$(O)=CHR$ (I(O)+48+7*(I(O)>9))
7100 NEXT O
7500 DIM S(2): DIM S$(2)
7510 LET S(1)=INT (D/16)
7520 LET S(2)=INT (D-S(1)*16)
7530 FOR F=1 TO 2
7540 LET S$(F)=CHR (S(F)+48+7*(S(F)>9))
7550 NEXT F
7560 RETURN
9999 SAVE "PROG+" LINE 10: SAVE "PROG+"CODE 32000,500:
     SAVE "MONITOR "CODE 61000,4535: STOP
```

```
7D00 3E 00 D3 EF DB 1F 32 0F
7D08 7D 00 3E 00 D3 EF C9 00
7D10 00 00 00 00 00 00 00 00
7D18 00 00 00 00 00 00 00 00
7D20 3E 80 D3 7F 7E D3 1F 78
7D28 D3 3F 79 D3 5F 3E 00 D3
7D30 EF 3E 00 D3 EF 3E 55 32
7D38 18 7D 32 18 7D 32 18 7D
7D40 3D 20 F4 3E 00 D3 EF 3E
7D48 90 D3 7F 78 D3 3F 79 D3
7D50 5F 3E 00 D3 EF DB 1F 32
```

```
7D58 5F 7D 3E 00 D3 EF 3E 00
7D60 BE C9 00 00 00 00 00 00
7D68 00 00 00 00 00 00 00 00
7D70 C5 D5 E5 F5 01 00 00 11
7D78 00 00 21 00 00 3E 90 D3
7D80 7F 78 D3 3F 79 D3 5F CD
7D88 00 7D 78 AA 20 09 79 AB
7D90 20 05 F1 E1 D1 C1 C9 03
7D98 23 18 E6 00 00 00 00 00
7DA0 00 00 00 00 00 00 00 00
7DA8 00 00 00 00 00 00 00 00
7DB0 C5 D5 E5 F5 3E 90 D3 7F
7DB8 3E 00 32 2F 7E 01 00 00
7DC0 11 00 00 21 00 00 78 D3
7DC8 3F 79 D3 5F CD 00 7D 3A
7DD0 0F 7D 00 00 28 2B 22 FF
7DD8 7D ED 53 FC 7D ED 43 F9
7DE0 7D F1 E1 D1 C1 C9 00 00
7DE8 00 00 00 00 00 00 00 00
7DF0 C5 D5 E5 F5 3E 90 D3 7F
7DF8 01 00 00 11 00 00 21 00
7E00 00 78 AA 20 1B 79 AB 20
7E08 17 3E FF 32 2F 7E F1 E1
7E10 D1 C1 C9 00 00 00 00 00
7E18 00 00 00 00 00 00 00 00
7E20 03 23 18 A2 00 00 00 00
7E28 00 00 00 00 00 00 00 00
7E30 C5 D5 E5 F5 3E 00 32 CF
7E38 7E 01 00 00 11 00 00 7A
7E40 32 F1 40 32 F1 41 7B 32
7E48 F2 40 32 F2 41 21 00 00
7E50 78 32 F1 43 32 F1 44 79
7E58 32 F2 43 32 F2 44 3E 33
7E60 3D 32 5F 7E 32 EC 43 32
7E68 EC 44 20 2C 22 90 7E ED
7E70 53 8D 7E ED 43 8A 7E F1
7E78 E1 D1 C1 C9 00 00 00 00
7E80 C5 D5 E5 F5 3E 33 32 5F
7E88 7E 01 00 00 11 00 00 21
7E90 00 00 18 16 00 00 00 00
7E98 CD 20 7D 20 C1 3E FF 32
7EA0 36 7D CD 20 7D 3E 55 32
7EA8 36 7D 78 AA 20 12 79 AB
7EB0 20 0E 3E FF 32 CF 7E F1
7EB8 E1 D1 C1 C9 00 00 00 00
7EC0 03 23 3E 33 32 5F 7E C3
7EC8 50 7E 00 00 00 00 00 00
```

ПРИМЕРЫ ИГРОВЫХ ПРОГРАММ ПЭВМ "СПЕКТРУМ"

Ниже приведены три игровые программы написанные на языке Бейсик [22].

Первая программа (программа № 1) моделирует вбрасывание монеты для игры в "китайку". После введения этой программы в компьютер необходимо запустить ее на выполнение, а затем нажать пять раз клавишу "ENTER" для получения двух гексограмм. Просмотрев "китайскую книгу измерений", играющий должен оценить глубину параллелей между ней и своей собственной жизнью. Нажав шестой раз клавишу "ENTER", можно обнулить программу, что избавит играющего от легкомысленного использования результатов.

Вторая программа (программа № 2) – игра "Ящеры", в которой компьютер отгадывает название животного, задуманного играющим. При этом компьютер задает вопросы, а играющий отвечает только "да" или "нет". Если компьютер не был ранее "знаком" с таким животным, то он "попросит" задать ему наводящие вопросы или "предложит" ему новое название.

Следующая программа (программа № 3) – это игра в слова. Первый игрок вводит слово, а второй отгадывает.

Программа № 1

```
5 RANDOMIZE
10 FOR M=1 TO 6: REM FOR 6 THROWS
20 LET C=0: REM INITIALIZE COIN TOTAL TO 0
30 FOR N=1 TO 3: REM FOR 3 COINS
40 LET C=C+2+INT(2*RND)
50 NEXT N
60 PRINT "   ";
70 FOR N=1 TO 2: REM 1ST FOR THE THROWN HEXAGRAM,
                      2ND FOR THE CHANGES
80 PRINT "---";
90 IF C=7 THEN PRINT "-";
100 IF C=8 THEN PRINT " ";
110 IF C=6 THEN PRINT "X";: LET C=7
120 IF C=9 THEN PRINT "0";: LET C=8
130 PRINT "---"
140 NEXT N
150 PRINT
160 INPUT A$
170 NEXT M: NEW
```

```
5 REM PANGOLINS
10 LET NQ=100: REM NUMBER OF QUESTIONS AND ANIMALS
15 DIM Q*(NQ,50): DIM A(NQ,2): DIM R*(1)
20 LET QF=8
30 FOR N=1 TO QF/2-1
40 READ Q*(N): READ A(N,1): READ A(N,2)
50 NEXT N
60 FOR N=N TO QF-1
70 READ Q*(N): NEXT N
100 REM START PLAYNG
110 PRINT "THINK OF AN ANIMAL.","PRESS ANY KEY TO
      CONTINUE."
120 PAUSE 0
130 LET C=1: REM START WITH 1ST QUESTION
140 IF A(C,1)=0 THEN GO TO 300
150 LET P*=Q*(C): GO SUB 910
160 PRINT "?": GO SUB 1000
170 LET IN=1: IF R*= "Y" THEN GO TO 210
180 IF R*= "Y" THEN GO TO 210
190 LET IN=2: IF R*= "N" THEN GO TO 210
200 IF R*<> "N" THEN GO TO 150
210 LET C=A(C,IN): GO TO 140
300 REM ANIMAL
310 PRINT "ARE YOU THINKING OF"
320 LET P*=Q*(C): GO SUB 900: PRINT "?"
330 GO SUB 1000
340 IF R*= "Y" THEN GO TO 400
350 IF R*= "Y" THEN GO TO 400
360 IF R*= "N" THEN GO TO 500
370 IF R*= "N" THEN GO TO 500
380 PRINT "ANSWER ME PROPERLY WHEN I'M" , "TALKING TO
        YOU." : GO TO 300
400 REM QUESSED IT
410 PRINT "I THOUQHT AS MUCH." : GO TO 800
500 REM NEW ANIMAL
510 IF QF>NQ-1 THEN PRINT "I'M SURE YOUR ANIMAL IS
    VERY" , "INTERESTING, BUT I DON'T HAVE" , "ROOM FOR
    IT JUST NOW." : GO TO 800
520 LET Q*(QF)=Q*(C): REM MOVE OLD ANIMAL
530 PRINT "WHAT IS IT, THEN?" : INPUT Q*(QF+1)
540 PRINT "TELL ME A QUESTION WHICH DIST-" , "INQUISHES
    BETWEEN"
550 LET P*=Q*(QF): GO SUB 900: PRINT "AND"
560 LET P*=Q*(QF+1): GO SUB 900: PRINT " "
570 INPUT S*: LET B=LEN S*
580 IF S*(B)= "?" THEN LET B=B-1
590 LET Q*(C)=S*(TO B): REM INSERT QUESTION
600 PRINT "WHAT IS THE ANSWER FOR"
610 LET P*=Q*(QF+1): GO SUB 900: PRINT "?"
620 GO SUB 1000
630 LET IN=1:LET IO=2: REM ANSWERS FOR NEW AND OLD
    ANIMALS
640 IF R*= "Y" THEN GO TO 700
```

```
650 IF R*= "Y" THEN GO TO 700
660 LET IN=2: LET IO=1
670 IF R*= "N" THEN GO TO 700
680 IF R*= "N" THEN GO TO 700
690 PRINT "THAT'S NO GOOD." :GO TO 600
700 REM UPDATE ANSWERS
710 LET A(C,IN)=QF+1: LET A(C,IO)=QF
720 LET QF=QF+2: REM NEXT FREE ANIMAL SPASE
730 PRINT "THAT FOOLED ME."
800 REM ADAIN?
810 PRINT "DO YOU WANT ANOTHER GO?" :GO SUB 1000
820 IF R*= "Y" THEN GO TO 100
830 IF R*= "Y" THEN GO TO 100
840 STOP
900 REM PRINT WITHOUT TRAILING SPASES
905 PRINT "  ";
910 FOR N=50 TO 1 STEP -1
920 IF P*<>" " THEN GO TO 940
930 NEXT N
940 PRINT P*(TO N);: RETURN
1000 REM GET REPLY
1010 INPUT R*: IF R*=" " THEN RETURN
1020 LET R*=R*(1): RETURN
2000 REM INITIAL ANIMALS
2010 DATA "DOES LIVE IN THE SEA" ,4,2
2020 DATA "IS IT SKALY" ,3,5
2030 DATA "DOES IT EAT ANTS" ,6,7
2040 DATA  "A WHALE" , "A BLANCMANGE" , "A PANGOLIN" ,
          "AN ANT"
```

```
5 REM HANGMAN
10 REM SETUP SCREEN
20 INK 0: PAPER 7: CLS
30 LET X=240: GO SUB 1000: REM DRAW MAN
40 PLOT 238,128: DRAW 4,0: REM MOUTH
100 SETUP WORD
110 INPUT W$: REMWORD TO GUESS
120 LET B=LEN W$: LET V$= " "
130 FOR N=2 TO W: LET V$=V$+ " "
140 NEXT N: REM V$=WORD GUESSED SO FAR
150 LET C=0: LET D=0: REM GUESS & MISTAKE COUNTS
160 FOR N=0 TO B-1
170 PRINT AT 20,N; "-" ;
180 NEXT N: REM WRITE-'S INSTEAD OF LETTERS
200 INPUT "GUESS A LETTER: " ;G$
210 IF G$= "" THEN GO TO 200
220 LET G$=G$(1): REM 1ST LETTER ONLY
230 PRINT AT 0,C;G$
240 LET C=C+1: LET U$=V$
250 FOR N=1 TO B: REM UPDATE GUESSED WORD
260 IF W$(N)=G$ THEN LET V$(N)=G$
270 NEXT N
280 PRINT AT 19,0;V$
290 IF V$=W$ THENGO TO 500: REM WORD GUESSED
300 IF V$<>U$ THEN GO TO 200: REM GUESSED WAS RIGHT
400 REM DRAW NEXT PART OF GALLOWS
410 IF D=3 THEN GO TO 600: REM HANGED
420 LET D=D+1
430 READ X0,Y0,X,Y
440 PLOT X0,Y0: DRAW X,Y
450 GO TO 200
500 REM FREE MAN
510 OVER 1: REM RUB OUT MAN
520 LET X=240: GO SUB 1000
530 PLOT 238,128: DRAW 4,0: REM MOUTH
540 OVER 0: REM REDRAW MAN
550 LET X=146: GO SUB 1000
560 RLOT 143,129: DRAW 6,0,PI 2: REM SMILE
570 GO TO 800
600 REM HANG MAN
610 OVER 1: REM RUB OUT FLOOR
620 PLOT 255,65: DRAW -48,0
630 DRAW 0,-48: REM OPEN TRAPDOOR
640 PLOT 238,128: DRAW 4,0: REM DRAW OUT MOUTH
650 REM MOVE LIMBS
655 REM ARMS
660 PLOT 255,117: DRAW -15,-15: DRAW -15,15
670 OVER 0
680 PLOT 236,81: DRAW 4,21: DRAW 4,-21
690 OVER 1: REM LEGS
700 PLOT 255,66: DRAW -15,15: DRAW -15,-15
710 OVER 0
720 PLOT 236,60: DRAW 4,21: DRAW 4,-21
730 PLBT 237,127: DRAW 6,0,PI/2: REM FROWN
```

```
740 PRINT AT 19,0;W$
800 INPUT "AGAIN?" ;A$
810 IF A$=  " THEN GO TO 850
820 LET A$=A$(1)
830 IF A$= "N" THEN STOP
840 IF A$(1)= "N" THEN STOP
850 RESTORE: GO TO 5
1000 REM DRAW MAN AT COLUMN X
1010 REM HEAD
1020 CIRCLE X,132,8
1030 PLOT X+4,134: PLOT X-4,134: PLOT X,131
1040 REM BODY
1050 PLOT X,123: DRAW 0,-20
1055 PLOT X,101: DRAW 0,-19
1060 REM LEGS
1070 PLOT X-15,66: DRAW 15,15: DRAW 15,-15
1080 REM ARMS
1090 PLOT X-15,117: DRAW 15,-15: DRAW 15,15
1100 RETURN
2000 DATA 120,65,135,0,184,65,0,91
2010 DATA 168,65,16,16,184,81,16,-16
2020 DATA 184,156,68,0,184,140,16,16
2030 DATA 204,156,-20,-20,240,156,0,-16
2020 DATA 184,156,68,0,184,140,16,16
2030 DATA 202,156,-20,-20,240,156,0,-16
```

ЛИТЕРАТУРА

1. **Гилмор Ч.** Введение в микропроцессорную технику/Пер. с англ. М.: Мир, 1984. 334 с.
2. **Дьяконов В.П.** Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ. М.: Наука, 1987. 240 с.
3. **Кориков А.М., Сырямкин В.И., Титов В.С.** Корреляционные зрительные системы роботов. Томск: Радио и связь. Томское отделение, 1990. 264 с.
4. **Волков А.В.** Ваш помощник – компьютер // Моделист-конструктор. 1987. № 2. С. 19.
5. **Зеленко Г.В., Панов В.В., Попов С.Н.** Домашний компьютер. М. : Радио и связь, 1989. 144 с.
6. **Косенков С.М., Полоскин А.Н., Счепицкий З.А. и др.** Бытовая персональная микроЭВМ "Электроника БК-0010"/ Микропроцессорные средства и системы. 1985. № 1. С. 22.
7. **МикроЭВМ.** Практ. пособие. В 8 кн. / Под ред. Л.Н. Преснухина. Кн. 8. МикроЭВМ в учебных заведениях / Г.И. Фролов, В.А. Шахнов, Н.А. Смирнов. М.: Высш. шк., 1988. 160 с.
8. **Персональная** ЭВМ "Партнер" // Радио. 1988. № 7. С. 66.
9. **Сугоняко В., Сафронов В., Коненков К.** Персональный радиолюбительский компьютер "Орион-128" // Радио. 1990. № 1. С. 37–43.
10. **Микрокалькулятор** "Электроника МК90": Руководство по эксплуатации. Минск, 1988. 106 с.
11. **Дудников Е.Е.** Персональные компьютеры. М.: Междунар. НИИ пробл. управл., 1988. 195 с.
12. **Tebbutt D.** Sinclair ZX81 // Perconal Computer World 1981. Vol. 4. No. 6. P. 67–70, 154.
13. **Wilson A.** Learnin the Lingo // Popular Computing Weekly 1984. 15–21 March. P. 16–17.
14. **Арсак Ж.** Программирование игр и головоломок / Пер с франц. М.: Наука, 1990. 224 с.
15. **Большие** возможности вашего "Спектрума": Учебно-методическое пособие для пользователей ПЭВМ "Спектрум". М.: НТК "Инфорком", 1990. 27 с.
16. **Sinclair** ZX81 Microcomputer. Hardware Catalogue. – Griffin and George, 1984.
17. **Surya.** Sinclair Expansion // Personal Computer World. 1984. Vol. 7. No. 1. P. 132.
18. **Mann S.** The Mighty Microdrive // Personal Computer World. Vol. 6. No. 10. P. 184–187.



УВЛЕКАТЕЛЬНЫЕ ИГРЫ НА БЫТОВОМ КОМПЬЮТЕРЕ

Малое предприятие «Раско»